NQuery 1.0

User Guide

First Edition

John R. Osborne
NOAA/PMEL
7600 Sand Point Way NE
Seattle, WA 98115
tel: 206-525-4585
email: John.Osborne@noaa.gov

NQuery development has been supported by NOAA's HPCC and PRIDE programs.

TABLE OF CONTENTS

CHAPTER 1. INTRODUCTION

About the User Guide

The User Guide is an attempt to document NQuery features in an indexed and consistent manner. The first edition of the User Guide is based on NQuery 1.0.

NQuery was designed to adhere to the native look and feel of the various computer operating systems it is installed. All examples in the first edition of the User Guide are taken from the Mac OS X version of NQuery, and so reflect that particular user interface style.

Please email to John.Osborne@noaa.gov your suggestions for corrections, additions and improvements to the User Guide.

Introduction to NQuery

NQuery is a Java-based, network-enabled data-based query tool that loads pertinent subsets of multi-disciplinary, earth-science datasets into a temporary, on-the-fly relational database, performs local calculations, and then allow a scientist to construct sophisticated SQL queries. What distinguishes NQuery from other data selection tools is the ability to find a subset of a dataset from the values of measured parameters rather than just the spatial domain. NQuery currently works only with oceanographic profile data but will be extended to time-series data in future revisions.

NQuery computes summary statistics (e.g., average value, depth of maximum value, and depth of minimum value) for a profile from the observed parameters and from user-specified calculations (e.g., theta, sigma, and apparent oxygen utilization.) User-defined calculations such as mixed-layer depth and interpolation of a measured parameter to a standard level van also be specified. A simple two-step process takes the user from pre-selected data (e.g., from local data collections, or distributed data collections available through Dapper/OPeNDAP) to a set of data files selected from the computed summary statistics. First, the user determines what observed and computed variables will be used to construct the database, and therefore are available for subsequent queries. This selection is accomplished via a simple graphical interface that lists all available variables. Additional dialogs allow selection and configuration of a large set of computed variables. A single click then begins the process of ingesting the data files, computing summary statistics for both the observed values and user-specified calculated values, and building and populating a MySQL database. Second, after the database has been populated with the requested data, the user can create a SQL query using a second graphical user interface.

The user can build simple to fairly complex queries by using either the graphical interface, or entering an SQL statement by hand. Once a satisfactory query is constructed, it is executed by the database, resulting in a list of files that satisfies the query. The scientist can then use these files in other research tools by exporting a “pointer file” that contains the locations of the actual data files.

Limitations of NQuery

NQuery is currently limited to profile data. NQuery can open data from multiple sources (pointer files, Dapper servers, and existing databases) into separate NQuery windows. However, NQuery currently does not have the capability to combine datasets from different sources into one NQuery document window (and thus into one on-the-fly database).

Reporting Bugs and Technical Support

Repairing software problems ('bugs') is an ongoing effort. Also, some existing features may, on occasion, develop bugs in newer versions. Through months of testing for each new version, bugs are found and repaired. If something goes wrong or NQuery doesn't act as expected, it's possible you may have encountered an error. Please contact the developer if this occurs. Appendix D has some additional information about what to do if NQuery acts unexpectedly.

Technical support is handled via electronic mail and the Internet. Send all technical questions to John ('Oz') Osborne <mailto://John.Osborne@noaa.gov>. He will make every attempt to resolve your problem as expediently as possible. Include the following information in your email message:

- Brief description of problem
- Repeatability of problem (a list of steps that reproduce a problem is very valuable)
- Version of NQuery
- Version of Java Virtual Machine (in the small type above the sponsor logos in the *About NQuery* window)
- Operating system version
- Amount of RAM in machine
- Data set specific problem? (In which case be prepared to send the data file to Oz or URL of Dapper server)

NQuery 1.0 Features

NQuery is a software application for ingesting oceanographic profile data and building a database from values calculated form the observed values. Currently, NQuery is limited to oceanographic profile data.

Platforms

NQuery should run on the following Java-enabled platforms:
Windows 95, 98, ME, NT, 2000*, XP
Mac OS X 10.3 and 10.4*
Solaris

Linux
Other UNIX OSes with at least JDK 1.4.2 support

Note: NQuery has not been tested on a wide variety of UNIX platforms or extensively on all flavors of Windows. NQuery has been used extensively with the operating systems marked with an asterisk (*).

Input/Output

EPIC XML Pointer Files (Input/Output)
EPIC Traditional Pointer Files (Output)
Argo and GTSPP Inventory files (Input)
Database Documents (Input/Output)
Dapper Servers (Input)

Calculations

Built-in Summary Statistic Calculations

- Minimum value of observed variable
- Maximum value of observed variable
- Depth of minimum value of observed variable
- Depth of maximum value of observed variable
- Maximum depth of non-missing value of observed variable
- Minimum depth of non-missing value of observed variable
- Average of non-missing value of observed variable
- Number of non-missing value of observed variable

Profile Scaler and Integral Calculations

- theta
- sigma-0
- sigma-1
- sigma-2
- sigma-3
- sigma-4
- sigma-n
- specific volume anomaly
- spiciness (Jackett and McDougall)
- sound velocity
- O_2 % saturation
- AOU
- NO
- PO
- Brunt-Vaisala frequency
- squared Brunt-Vaisala frequency
- squared Brunt-Vaisala frequency/g
- thermal expansion (alpha)
- alpha * dT/dz
- saline contraction (beta)
- beta * dS/dz
- neutral density (gamma—Jackett and McDougall)
- acoustic travel time
- net heat content
- geopotential anomaly

- potential energy anomaly

"Station" Calculations

- Mixed-layer depth based upon variable of choice and choice of difference method, surface method, or slope method; User-settable tolerance
- Integration of any variable between any two values of another variable (options include: use shallowest/deepest observation if surface outcrops or hits bottom if needed, compute weighted mean, interpolation of missing values and interpolation direction: top down, bottom up, or up/down)
- Interpolation of any variable onto surface value of any other variable; special interpolation to surface or bottom of ocean

CHAPTER 2. INSTALLING NQUERY & SUPPORT FILES

Mac OS X Installation Notes

1. Download or copy from the distribution CD the zipped NQuery disk image file.

2. Double click the zip file to unarchive it to a disk image file (.dmg).

3. Double click the .dmg file to mount the disk on the desk top.

4. Open the disk image and drag the NQuery folder to your Applications folder (or a folder of your choosing).

Mac OS X System Requirements

- OS X version 10.3.9 or higher
- 256 MB RAM (512 MB or more recommended).
- G3 class processor (G4/G5/Intel recommended).
- At least 100 MB hard-disk space.

Windows Installation Notes

1. If your machine already has JRE 1.4.2 (or JDK) already installed then download or copy from the distribution CD "NQuery w/o JVM", otherwise; download an installer that includes JVM called "NQuery & JVM".

2. Double click the installation file.

3. Follow the on-screen instructions to complete the installation.

Windows System Requirements

- Microsoft Windows 2000/XP.
- 256 MB RAM (512 MB or more recommended).
- At least 100 MB hard-disk space.

UNIX Installation Notes

1. Download a NQuery UNIX installer or copy from the distribution CD for either Linux, Solaris or generic UNIX. These installers do not contain an embedded Java Virtual Machine.

2. Run "chmod +x installer_name" on the command line (make it executable).

3. Start the installer by invoking it's name from a command line (from within an X session).

4. Follow on-screen instructions.

5. If you have more than one JVM installed on your machine, the installer will prompt you to select a one.

Unix System Requirements

- UNIX (Solaris/Linux/BSD etc.).
- 256 MB RAM (512 MB or more recommended).
- At least 100 MB hard-disk space.

CHAPTER 3: INSTALLING MYSQL

Where to get the Installer: As of 02/06, the latest version of MySQL is 5.0. Currently NQuery has only been tested with MYSQL 4.1.5. A good starting point to find the 4.1.x installer is **http://dev.mysql.com/downloads/** Follow the link for MySQL 4.1 (currently **http://dev.mysql.com/downloads/mysql/4.1.html**).

For all platforms, there is both a *standard* and *max* version available. NQuery was based on the standard version of the software. The max version has additional capabilities not required by NQuery.

Installation on Mac OS X

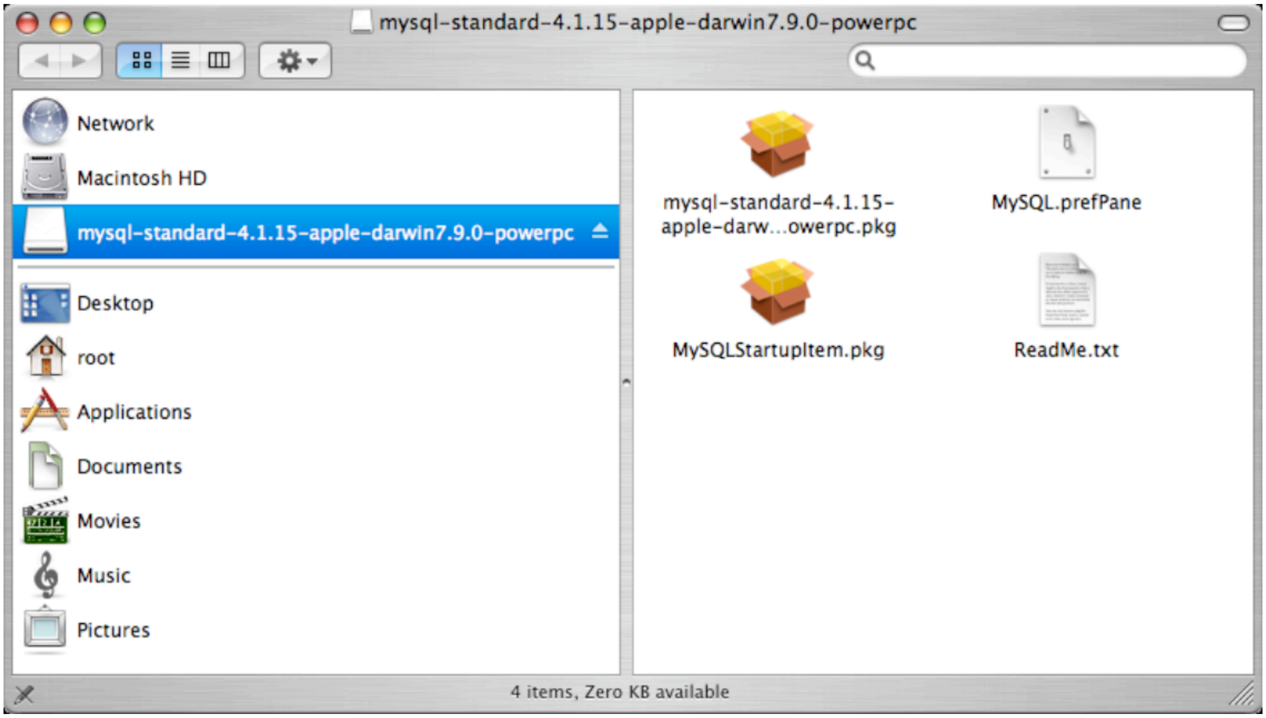
0) This section describes installing MySQL through a graphical wizard. If you would prefer to install MySQL using the command line interface, see the documentation at:

http://dev.mysql.com/doc/refman/4.1/en/mac-os-x-installation.html

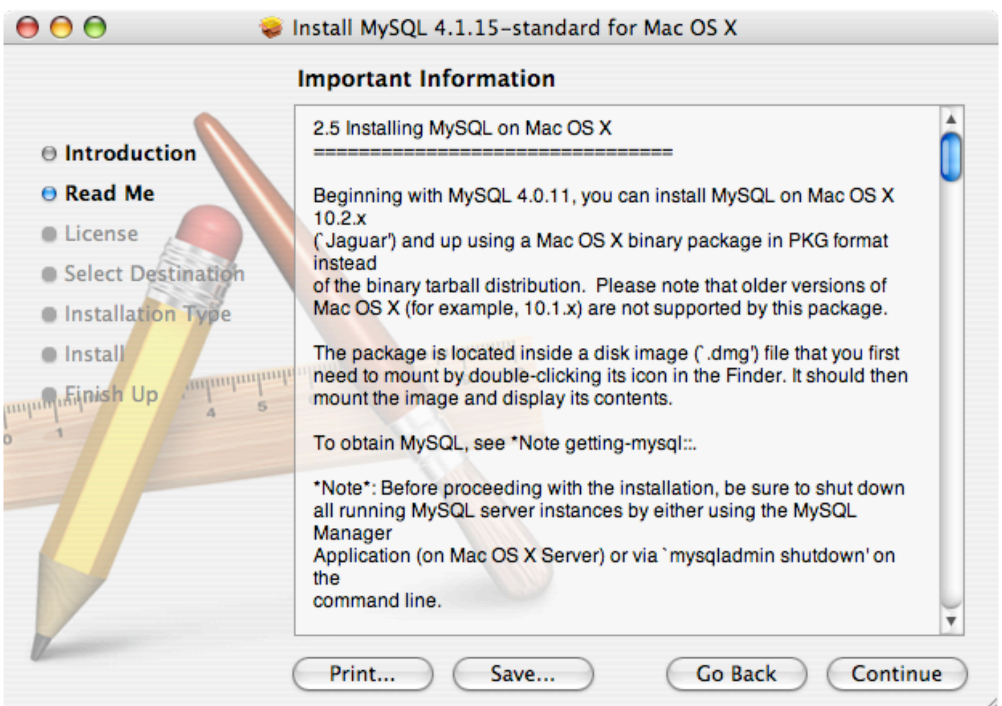
1) Download the MySQL installer package for Mac OS X. The installer will be packaged as a “disk image” file with a .dmg file extension. The disk image should automatically open on the desktop with a white external disk icon:

If not, double click the .dmg file to mount the disk image.

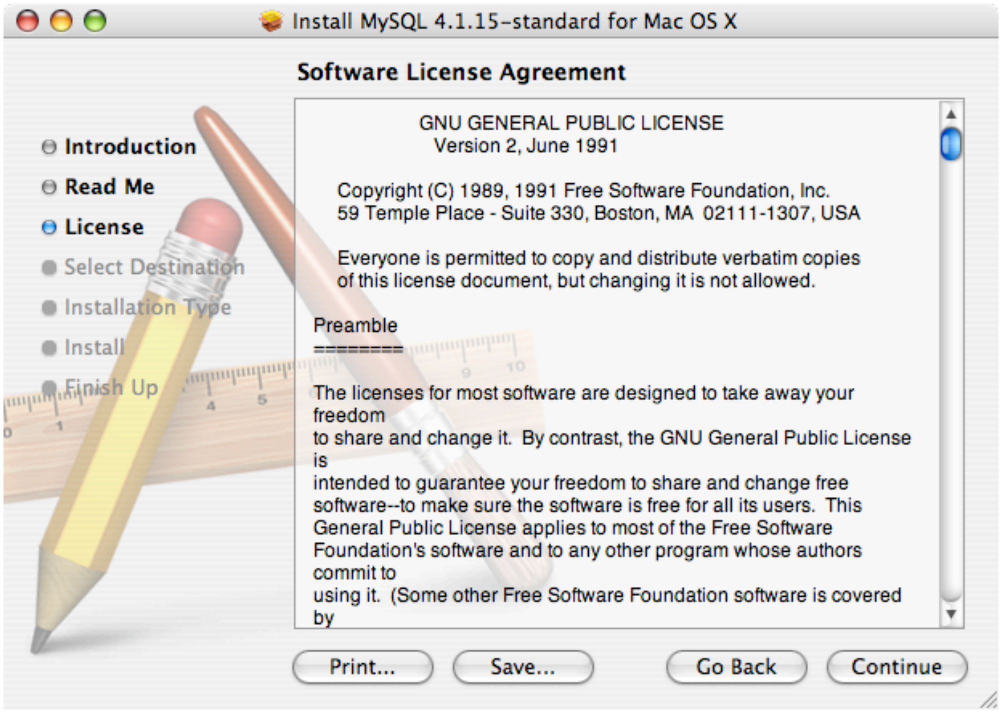
2) Open the disk image to reveal its contents. You can open the disk image into a new window by double-clicking the disk image icon or click on the disk image in the sidebar:

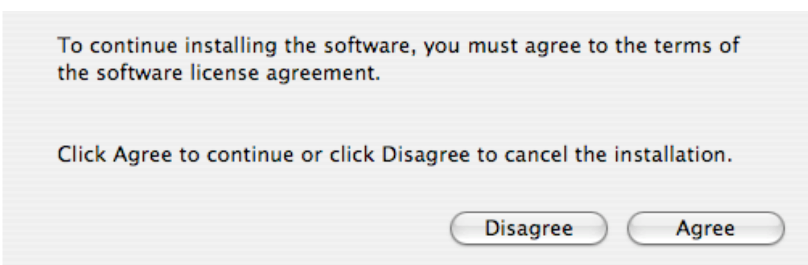


3) Double click on the MySQL installer package (a file with a .pkg extension). In this example, the installer package is called "mysql-standard-4.1.5-apple-darwin-powerpc.pkg." This will launch the MySQL installation assistant (or wizard):

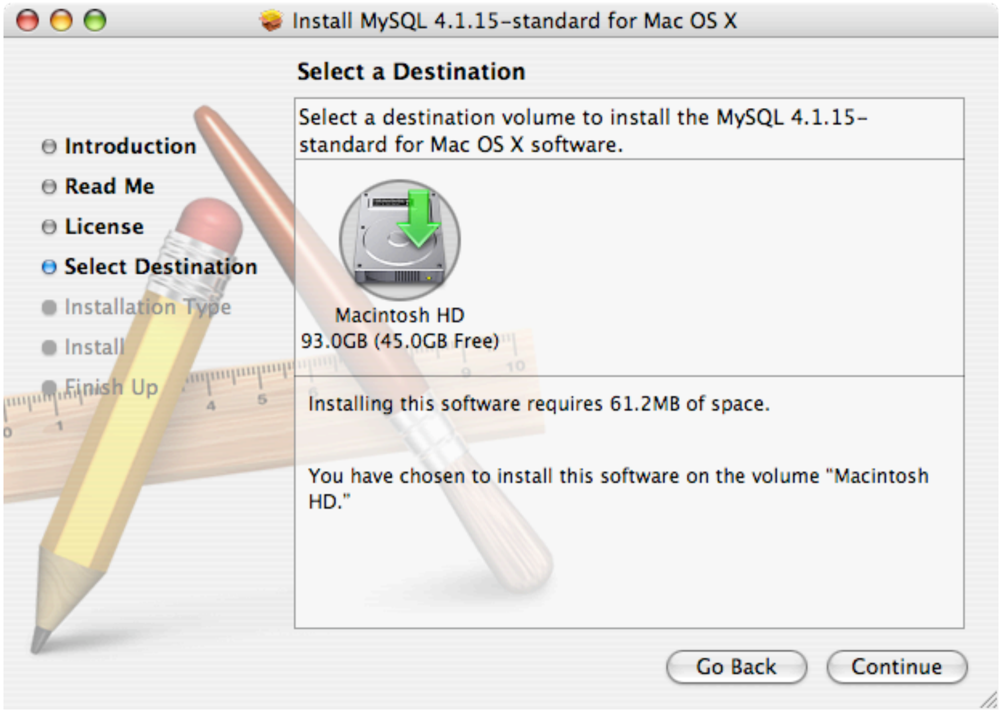


4) Click the *Continue* button to proceed through the installation process. Click the *Agree* button to accept the GPL agreement:



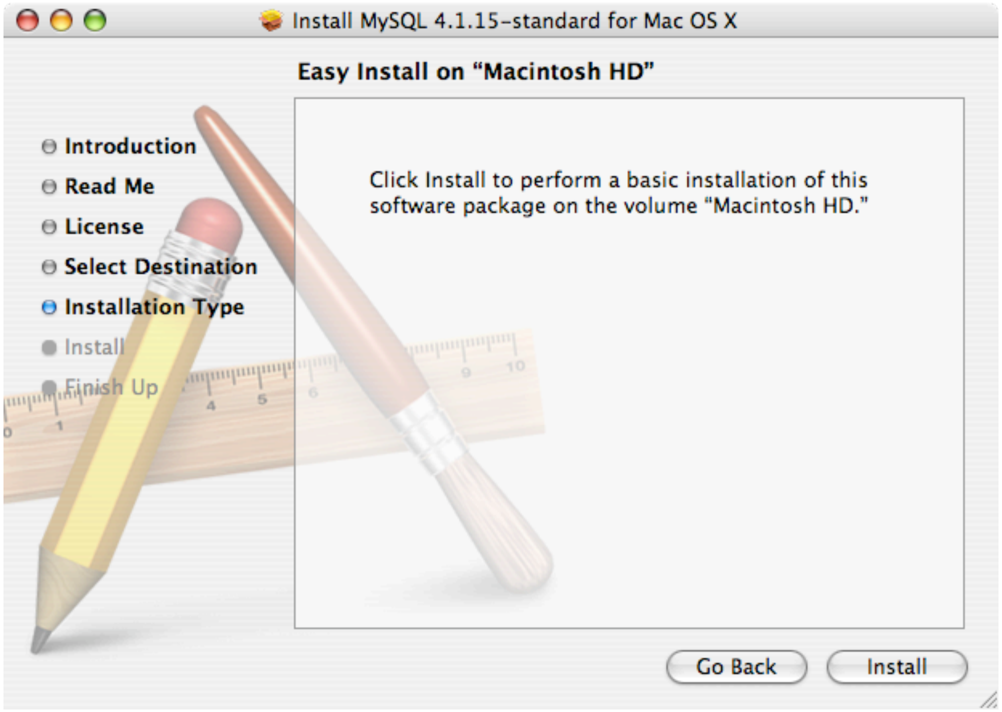


5) Select the destination volume to install MySQL:

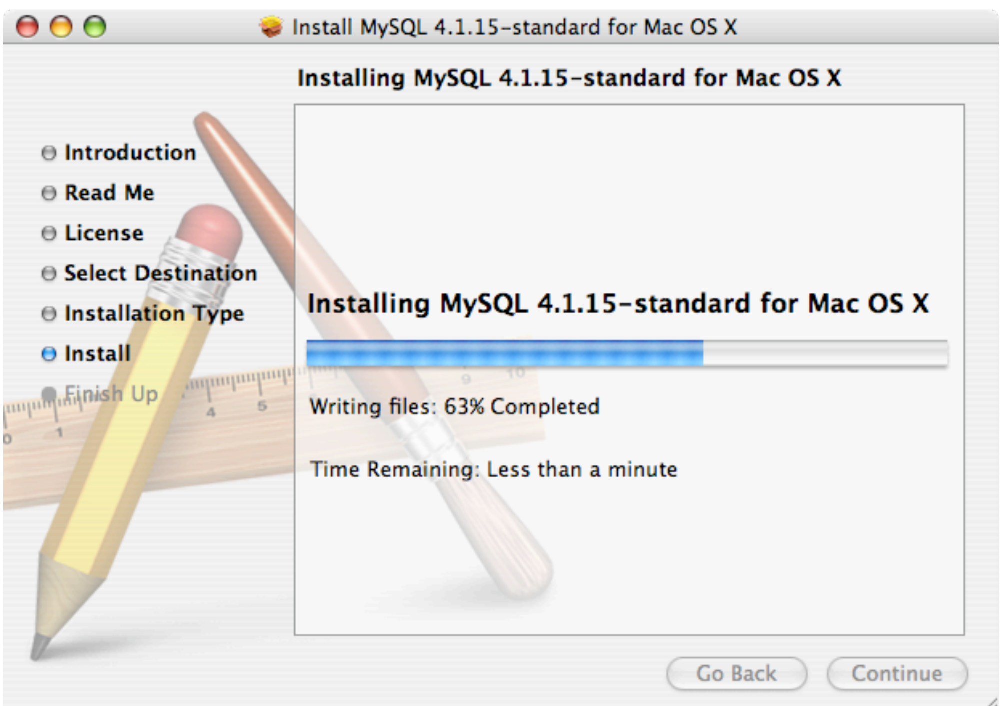


6) Enter an administrator's name and password and click *OK*:

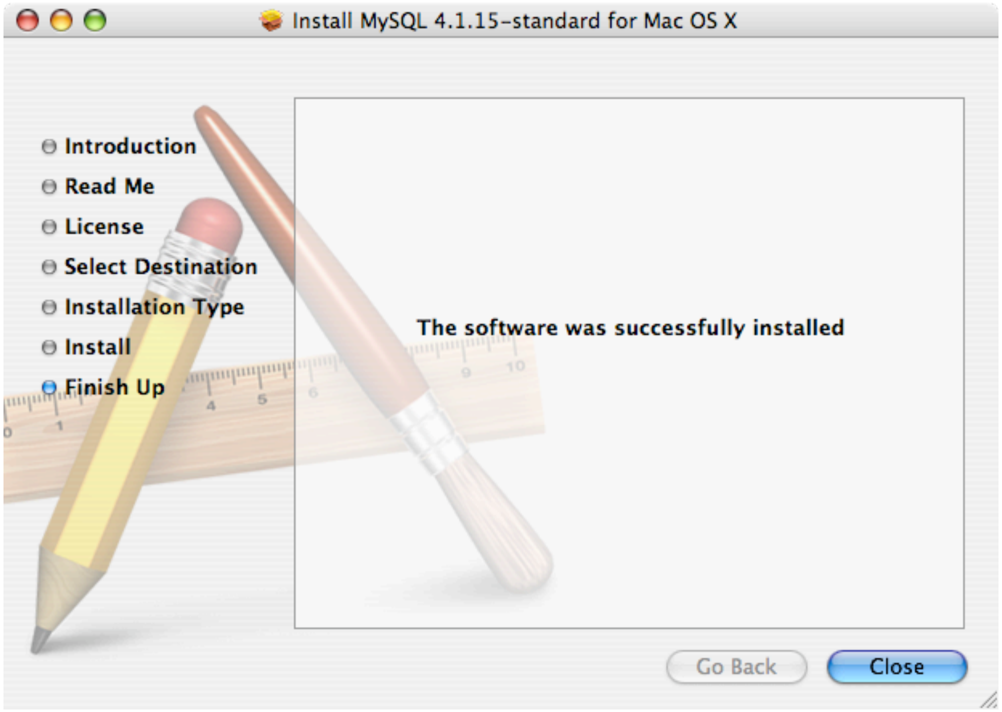
7) There is only one installation type for Mac OS X. Click the *Install* button to start installing MySQL on your Mac:



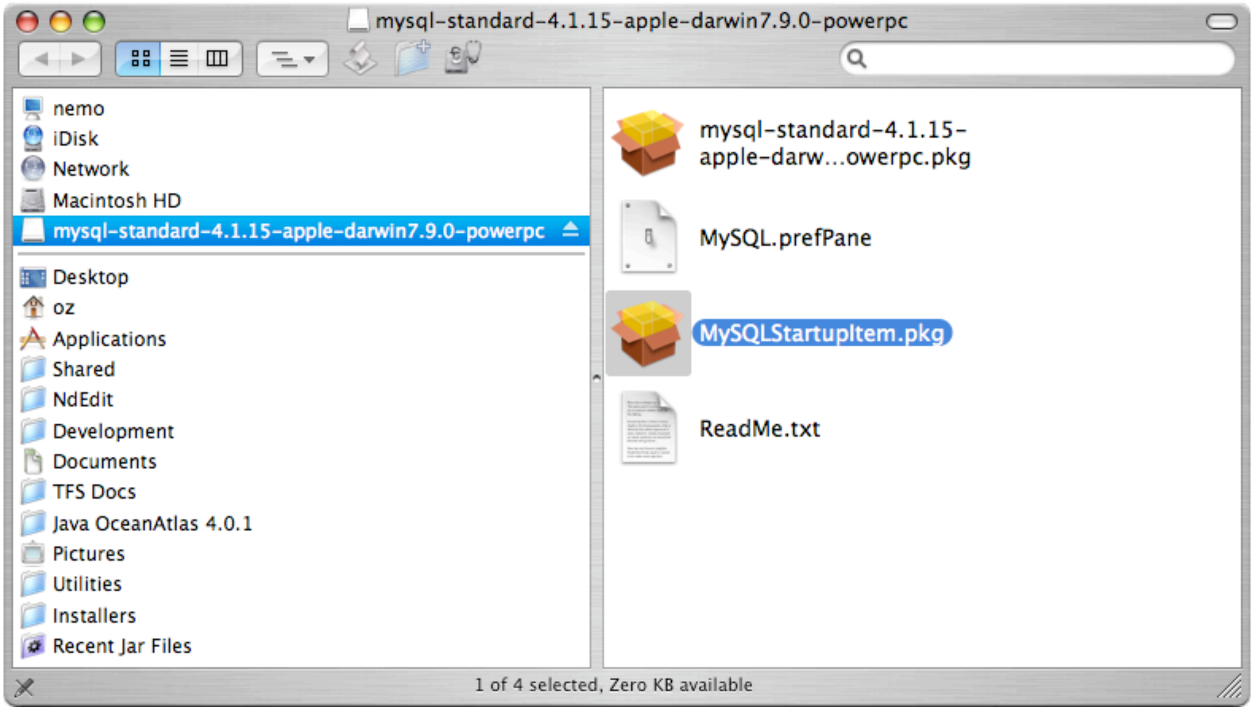
8) You will be informed of the installation progress:



9) You will be notified when the installation is complete. Click the *Close* button to quit the installer application:

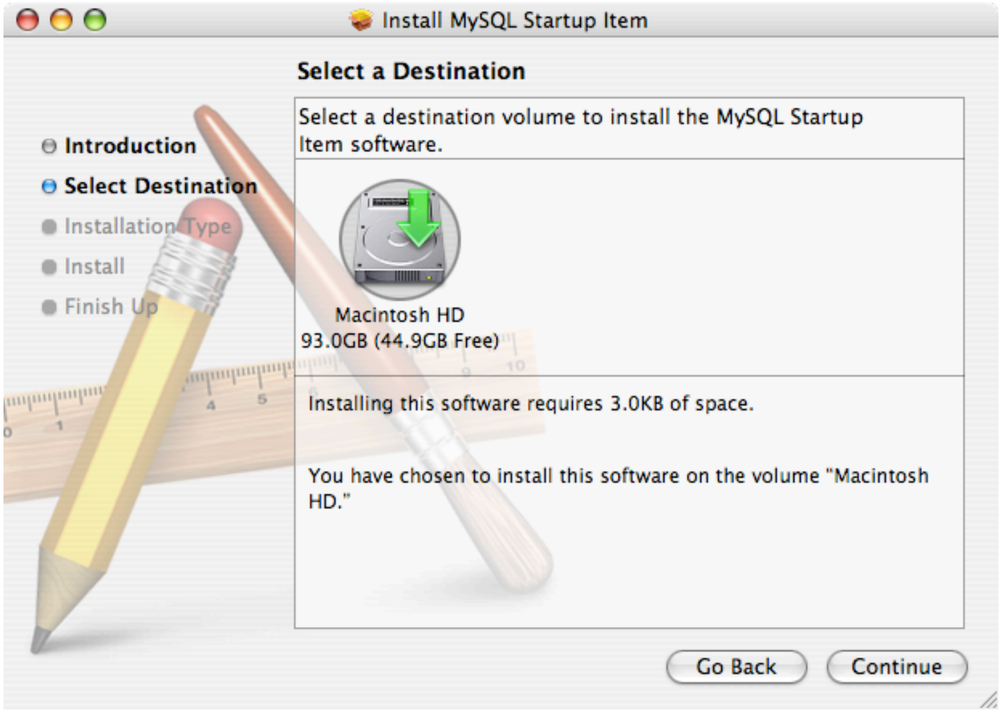


10) At this point MySQL is installed on your Mac but not actually running. MySQL can be started manually through the command line interface but installing the MySQL startup item is recommended for most users. In the MySQL installer disk image, double click the MySQLStartupItem.pkg:

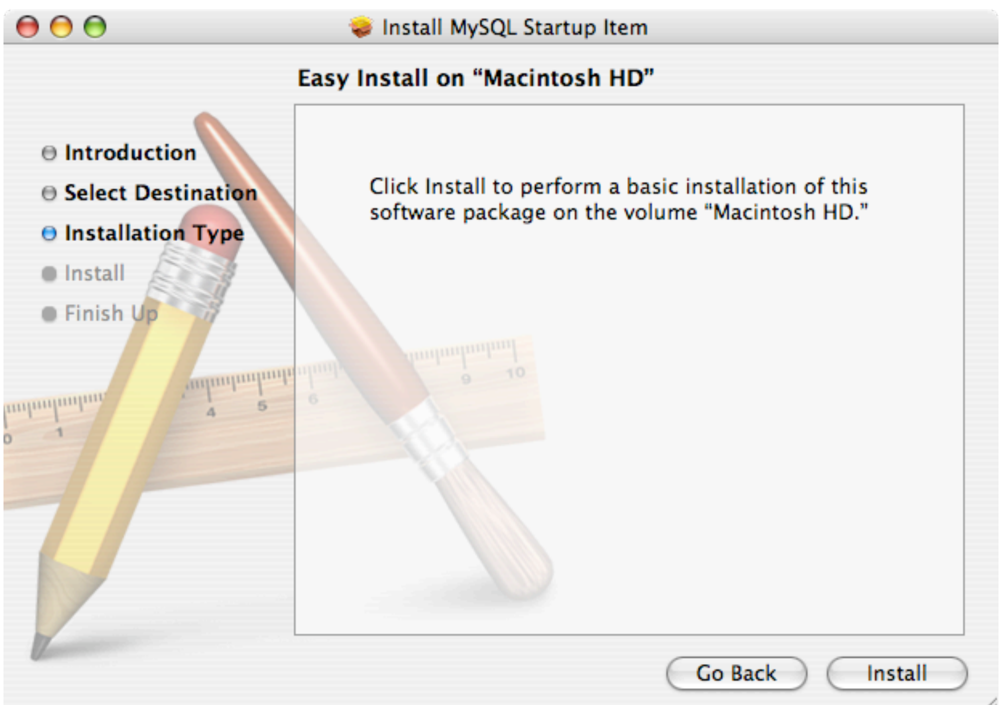


11) This installer will install a MySQL startup control panel in the System Preferences application. This preference pane makes it easy to start/stop the database server as well as view its current status. Follow the installation wizard:

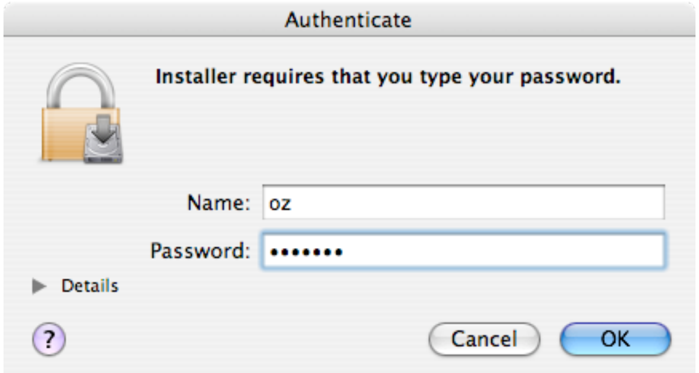
12) Select the same destination volume you installed MySQL on:



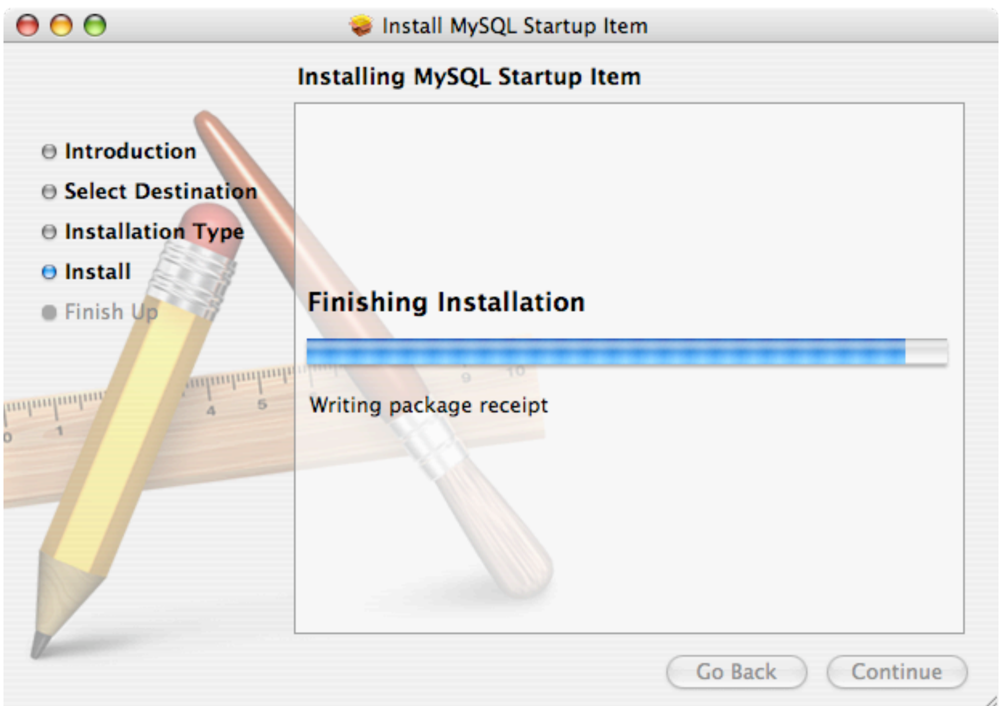
13) Click the *Continue* button:



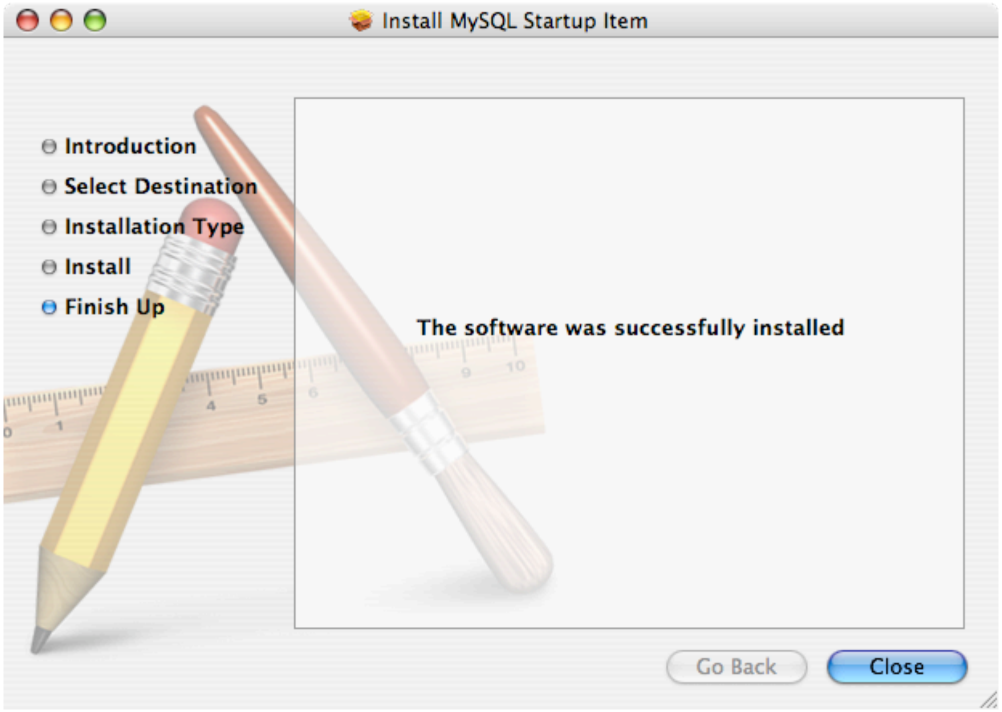
14) Provide an administrator's user name and password and click *OK*:



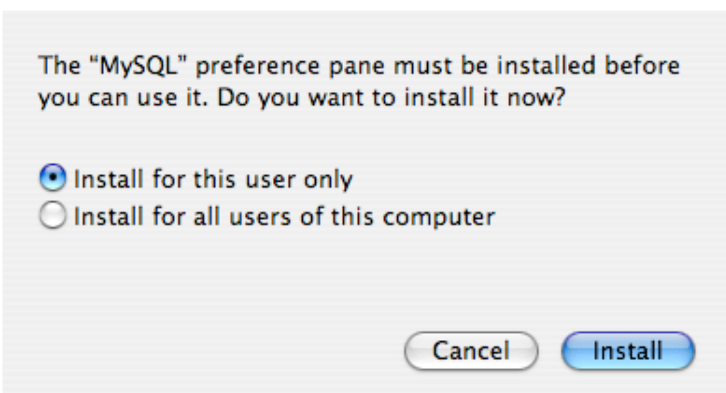
15) You will be informed of the installation progress:



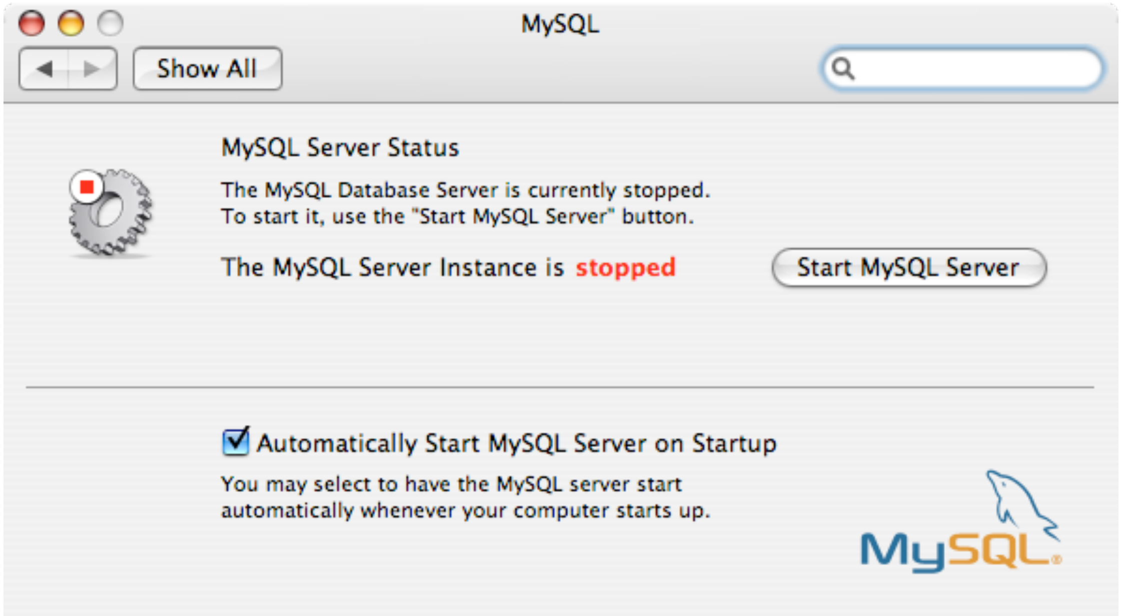
16) Click the *Close* button to quit the installer:



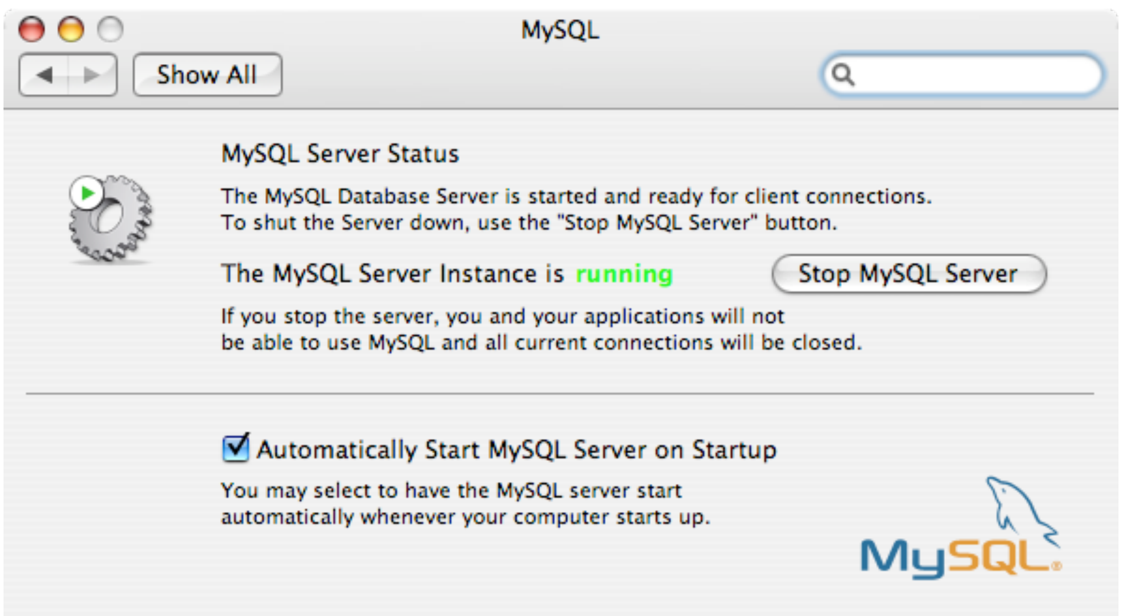
17) At this point, the startup item is installed on the disk but not activated as a System Preference. Double click the file: MySQL.prefPane found on the MySQL disk image to activate the preference pane. A bew dialog opens that allows you to chose to restrict this item to the current user or make it available to all users of your Mac:



18) Open System Preferences from the Apple menu or the dock. Click on the MySQL item in the *Other* panel of the System Preferences window. Initially, the MySQL server is "stopped:"



19) Click the *Start MySQL Server* button. Status should changed to “running.” It is recommended that you check *Automatically Start MySQL Server on Startup*:



Installation on Windows

Complete instructions can be found at:

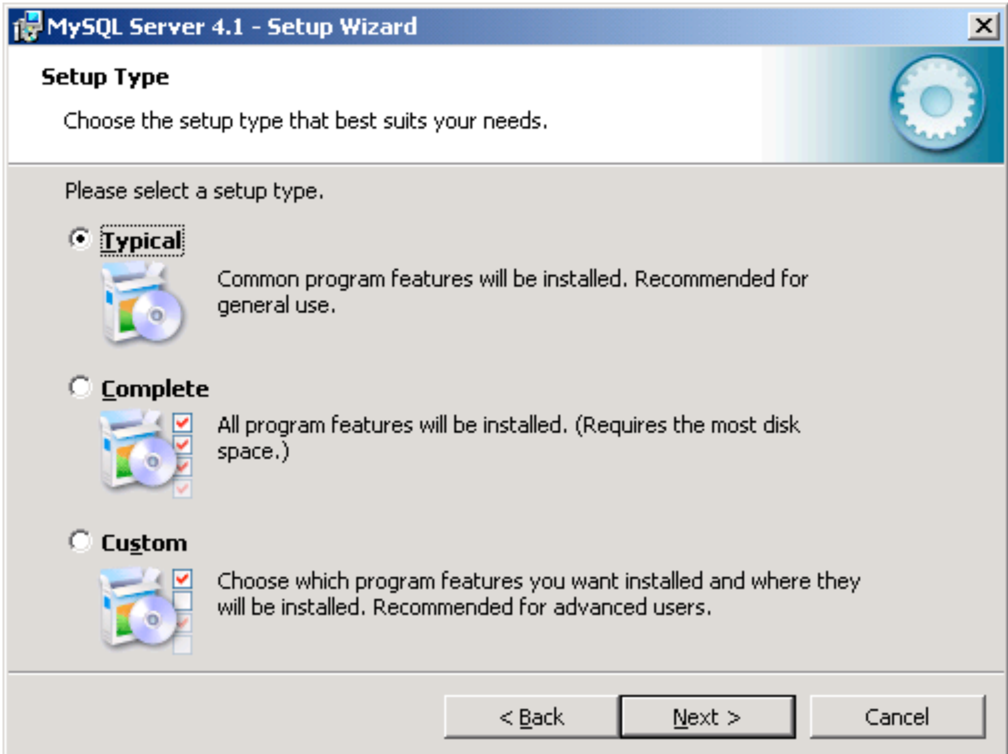
http://dev.mysql.com/doc/refman/4.1/en/windows-installation.html

1) Download the MYSQL installer package. For Windows, the installer comes in a zip file. Open the zip file and extract the actual installer to your disk.

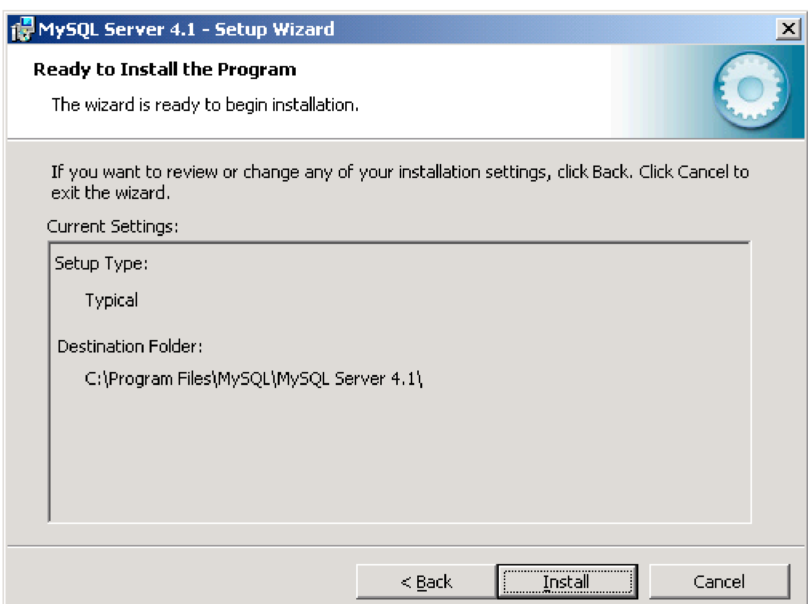
2) Launch the installer wizard and follow the instructions. Use the following screen images to set MySQL to a recommended configuration:

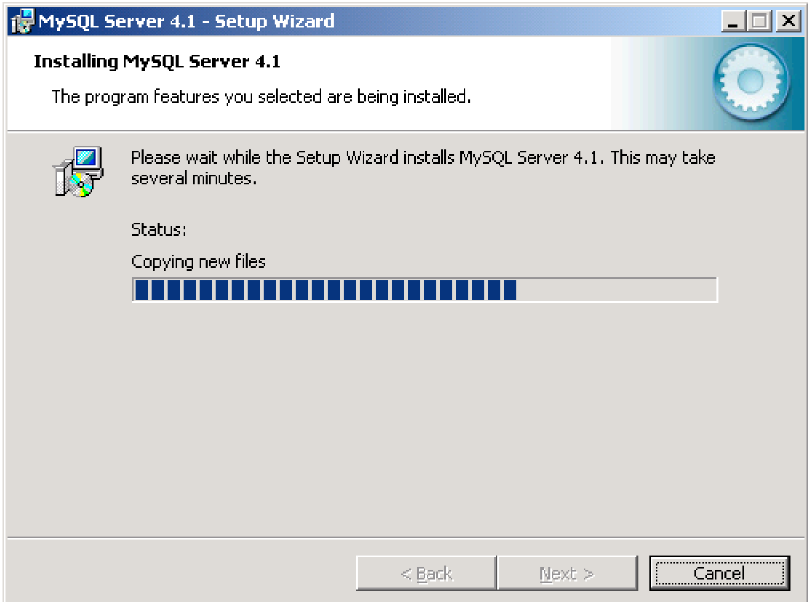
Click *Next* to continue.

3) The *Typical* setup type installs a useable MySQL system for NQuery. Other setup types are beyond the scope of this document.



4) The wizard is now ready to install MySQL. Click the *Install* button to begin installation. You will see a progress screen as the installation proceeds:

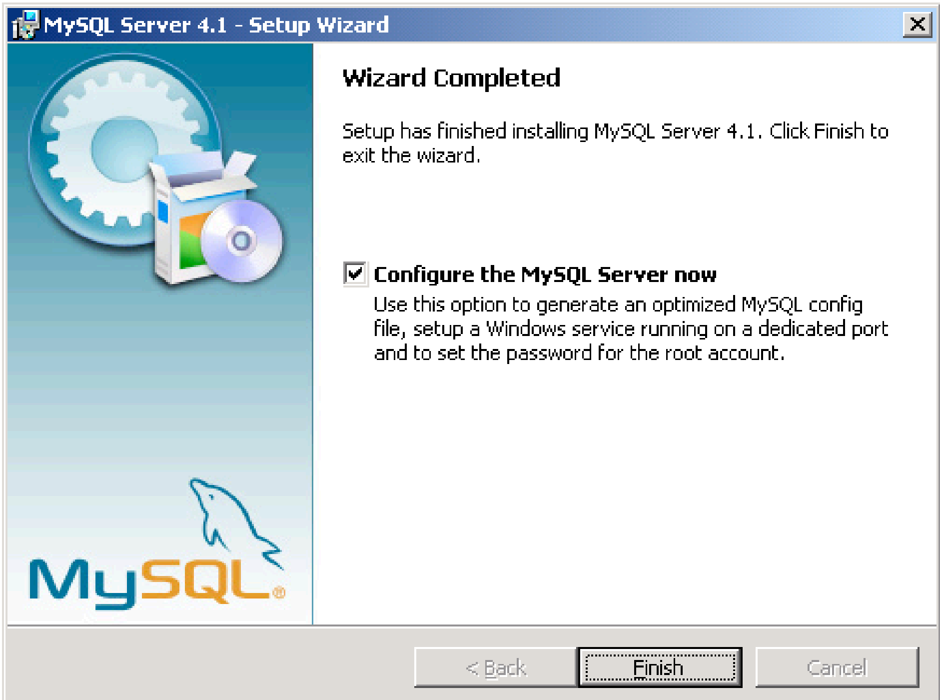




5) You may be prompted to set up an account at MySQL.com:

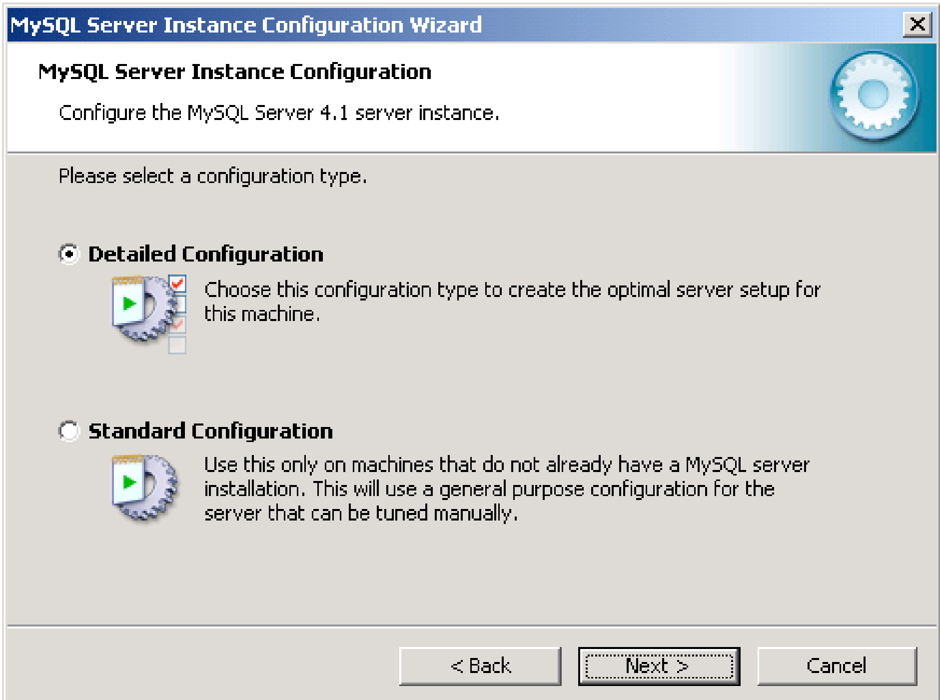
Note: Setting up an account at MySQL.com is beyond the scope of this document.

6) When the installation process is complete, you will be prompted to configure the server. We recommend that you follow the steps in the next section.

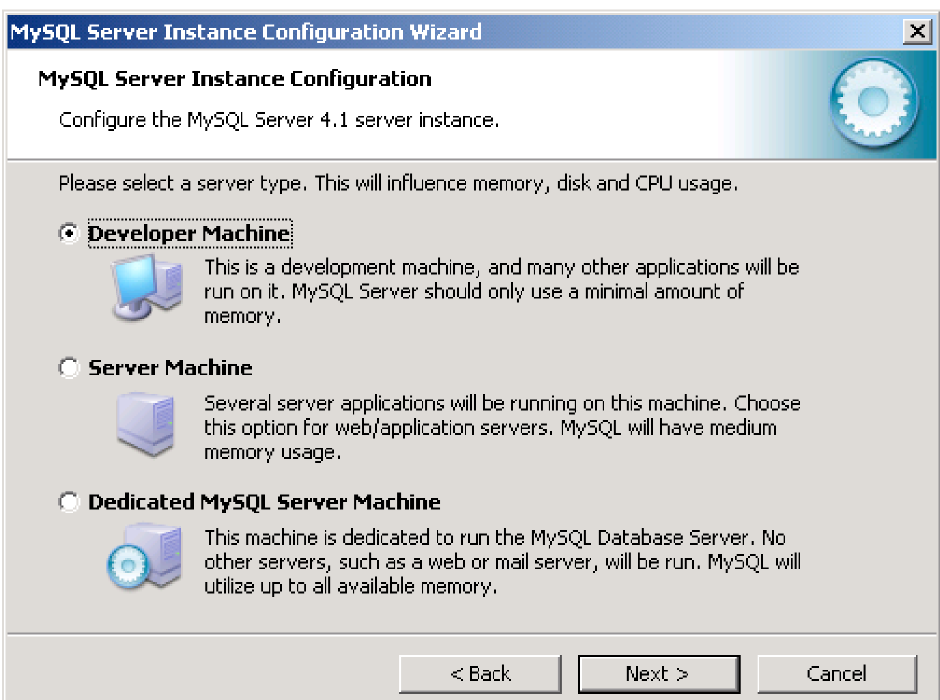


Configuring the MySQL Server on Windows

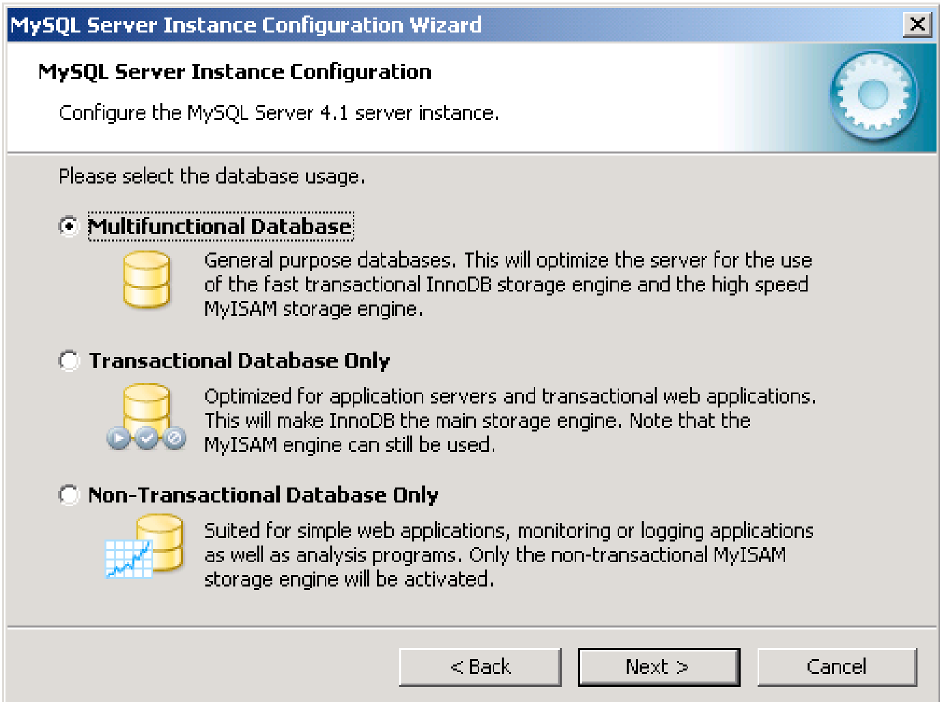
After installation has completed, you will be prompted to run the MySQL Instant Configuration Wizard. Unless you are comfortable with the command line interface of your operating system), we recommend using this wizard in the *Detailed Configuration* mode for initial configuration:



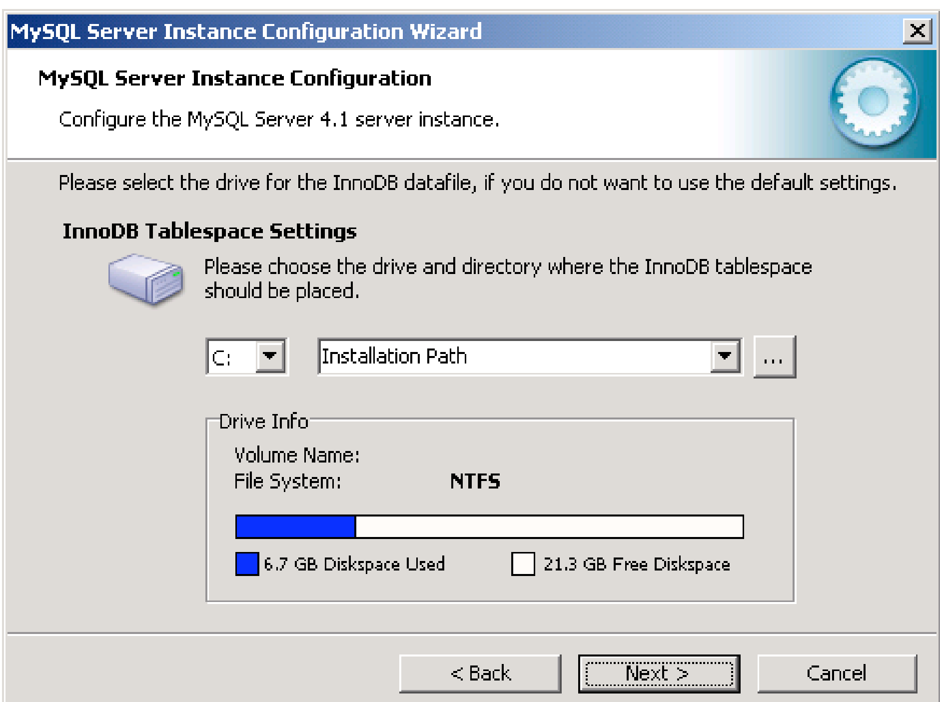
7) The *Developer Machine* option is best for a general-purpose desktop computer (other configurations are beyond the scope of this document):



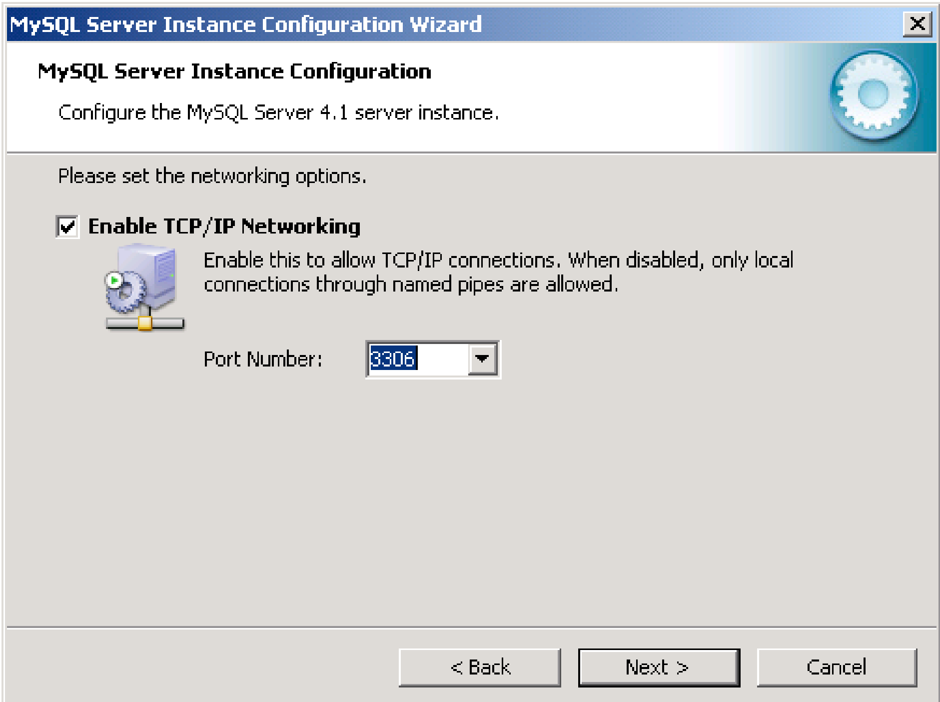
8) Select the *Multifunctional Database* option (other settings are beyond the scope of this document):



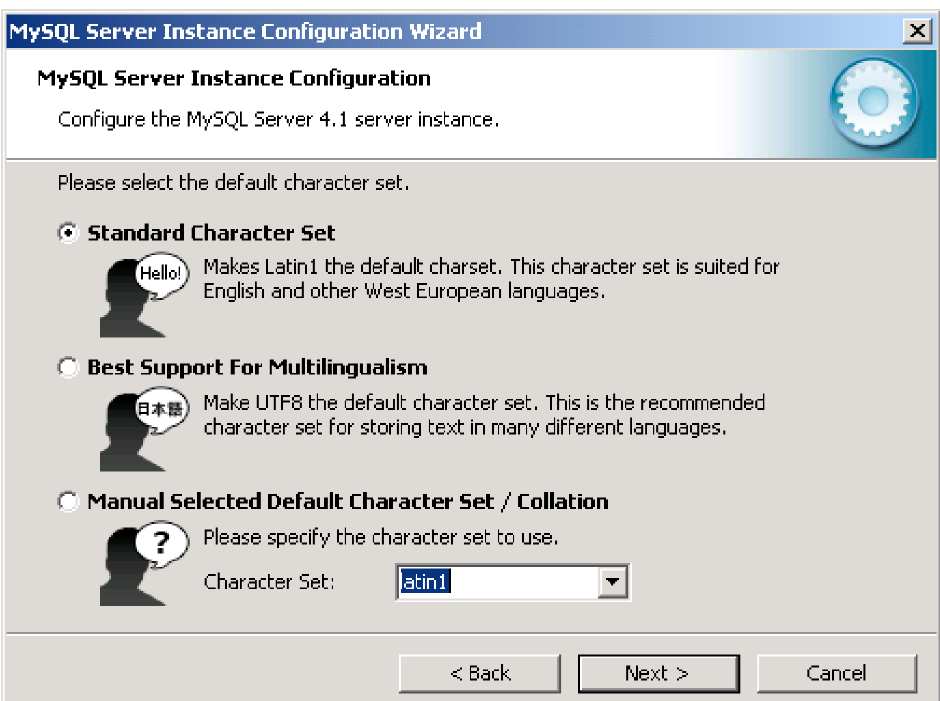
9) If disk space is tight on your PC, you can choose to store the database files on another disk or partition. Here we accept the default of using the main volume of the PC:



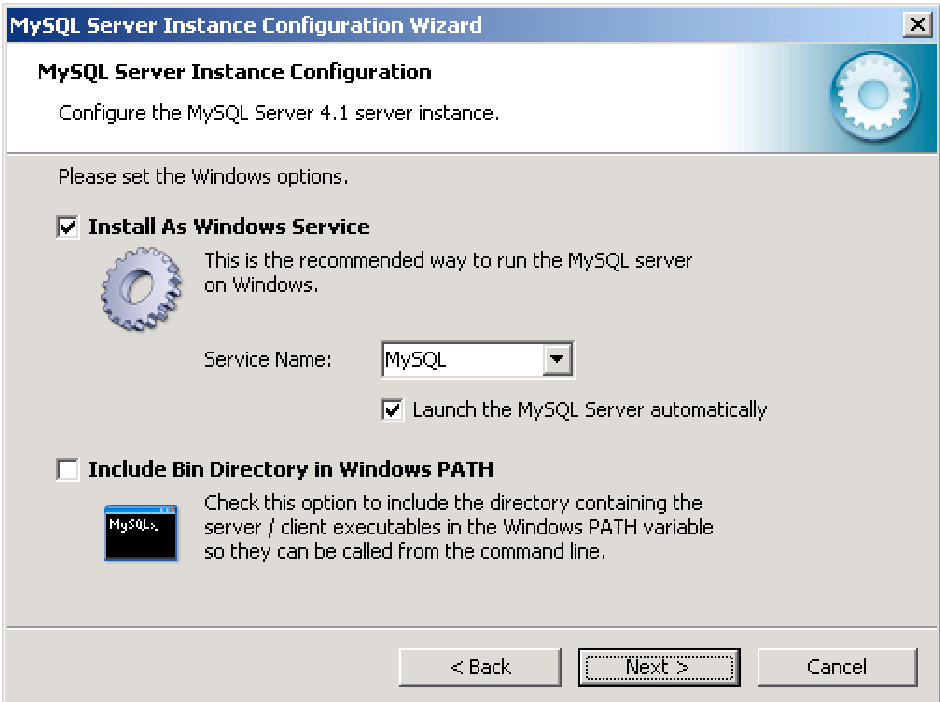
10) Unless you know how to use named pipes, enable *TCP/IP networking* on the default port number (3306):



11) You may wish to choose an alternative character set:

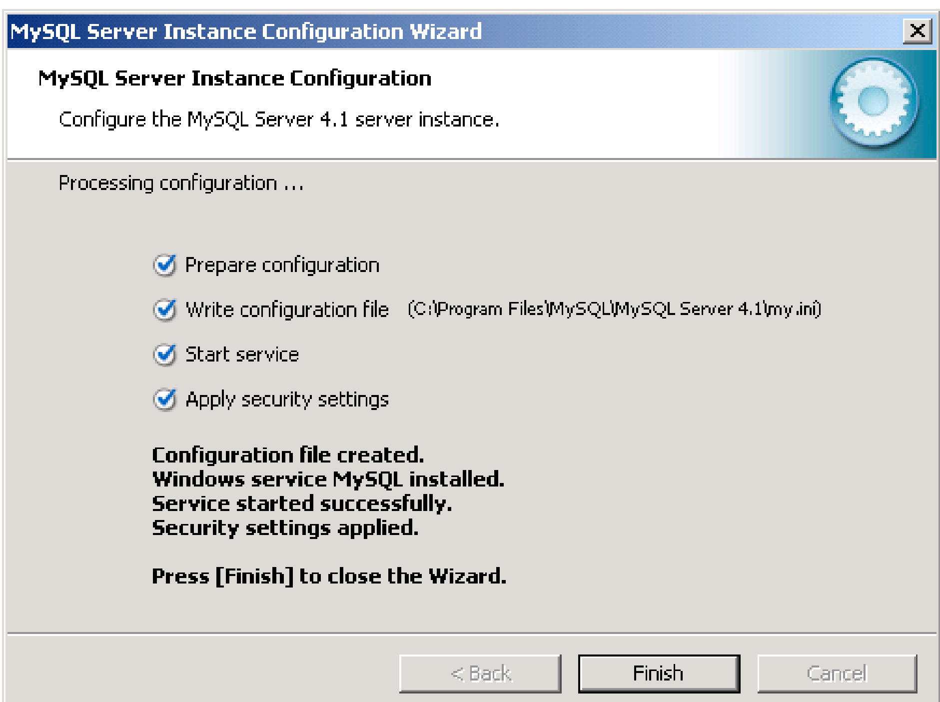


12) We highly recommend installing MySQL startup as a *Windows Service*. Make sure the *Launch the MySQL Server automatically* button is checked. Otherwise, you will be responsible for starting MySQL from the command line:



13) MySQL comes with one defined user with the name “root”. You may chose to use this user or you may setup other user names. If you do not have a separate MySQL management application, then new users will have to be added via the command line. We recommend that you assign a password to the root account whether you define other users or not:

14) If you see the screen below, then MySQL was successfully configured and you can now create databases in NQuery. Click the *Finish* button to begin using MySQL and NQuery.



Installation on Linux or UNIX

There are many different flavors of Linux and this document cannot describe them all. Complete instructions can be found at:

http://dev.mysql.com/doc/refman/4.1/en/linux-rpm.html

MySQL Initial Setup

Adding Users to MySQL Database Server

After MySQL is first installed it can be accessed by anyone on your system. The MySQL installation process typically adds one or two *root* accounts as well as two anonymous user accounts. The root and user accounts have full control over the server so it's important that the first thing you do is secure the root accounts by adding passwords. Note: the Windows installer, allows you to secure MySQL in the installation wizard. This section is devoted to Mac OS X and UNIX users.

Securing MySQL and Adding Users on Mac OS X

User's can be added to the MySQL server either through the command line interface or through third party, free or commercial graphical programs. One recommended commercial tool is *MySQL4X Manager* described at:

http://www.macosguru.com/macosguru3/products/mysql4xmanagergeneric.html

This manual will describe how to do this from the command line—it's beyond the scope of this manual to describe all the available graphical tools.

To see this process in excruciating detail, go to:

http://dev.mysql.com/doc/refman/4.1/en/default-privileges.html

Note: Any new users have to be added with all privileges so NQuery can create bew databades and delete old databases.

First it's useful to know a little about where MySQL is installed on your Mac. MySQL is installed into */usr/local/mysql*. However, *mysql* is actually a link to a different directory so the real installation directory will look more like: */usr/local/mysql-standard-4.1.15-apple-darwin7.9.0-powerpc*. This directory contains the following items:

```
drwxr-xr-x   19 root   wheel    646 Oct 10 03:59 .
drwxr-xr-x   12 root   wheel    408 Jan 18 04:00 ..
-rw-r--r--    1 root   wheel  19071 Oct  9 15:46 COPYING
-rw-r--r--    1 root   wheel   5712 Oct 10 03:54 EXCEPTIONS-CLIENT
-rw-r--r--    1 root   wheel   8307 Oct 10 03:54 INSTALL-BINARY
```

```
-rw-r--r--    1 root   wheel   1379 Oct  9 15:46 README
drwxr-xr-x   54 root   wheel   1836 Oct 10 03:59 bin
-rwxr-xr-x    1 root   wheel    801 Oct 10 03:58 configure
drwxr-x---   23 mysql  wheel    782 Jan 21 17:03 data
drwxr-xr-x    4 root   wheel    136 Oct 10 03:58 docs
drwxr-xr-x   61 root   wheel   2074 Oct 10 03:59 include
drwxr-xr-x   10 root   wheel    340 Oct 10 03:59 lib
drwxr-xr-x    3 root   wheel    102 Oct 10 03:58 man
drwxr-xr-x   12 root   wheel    408 Oct 10 03:59 mysql-test
drwxr-xr-x    3 root   wheel    102 Oct 10 03:59 scripts
drwxr-xr-x    5 root   wheel    170 Oct 10 03:59 share
drwxr-xr-x   31 root   wheel   1054 Oct 10 03:59 sql-bench
drwxr-xr-x   14 root   wheel    476 Oct 10 03:59 support-files
drwxr-xr-x   21 root   wheel    714 Oct 10 03:59 tests
```

The most useful directory here is the *bin* directory. It has these important files:

mysql: the command line interface to the actual database server
    http://dev.mysql.com/doc/refman/4.1/en/mysql.html
mysqladmin: client for performing administrative operations
    http://dev.mysql.com/doc/refman/4.1/en/mysqladmin.html
mysqld: the actual MySQL server
    http://dev.mysql.com/doc/refman/4.1/en/server-options.html
mysqlshow: Display database, table, and column information
    http://dev.mysql.com/doc/refman/4.1/en/mysqlshow.html

You will need to be an administrator to run the commands below. You will note that the commands are prefaced with “sudo”. This stands for “super user do”. You will be prompted for an administrative password to complete the command.

1) Verify that the MySQL server is running:

```
>sudo /usr/local/mysql/bin/mysqlshow
Password:

+-----------+
| Databases |
+-----------+
| mysql     |
| test      |
+-----------+
```

or:

```
> cd /user/local.mysql/bin
> sudo ./mysqladmin version
Password:
```

```
./mysqladmin  Ver 8.41 Distrib 4.1.15, for apple-darwin7.9.0 on
powerpc
Copyright (C) 2000 MySQL AB & MySQL Finland AB & TCX DataKonsult AB
This software comes with ABSOLUTELY NO WARRANTY. This is free
software,
and you are welcome to modify and redistribute it under the GPL
license

Server version          4.1.15-standard
Protocol version        10
Connection              Localhost via UNIX socket
UNIX socket             /tmp/mysql.sock
Uptime:                 3 days 22 hours 38 min 17 sec

Threads: 1  Questions: 14647  Slow queries: 0  Opens: 940  Flush
tables: 1  Open tables: 51  Queries per second avg: 0.043
```

Note in the above example we have moved into the mysql directory with the *cd* command. The "./" in subsequent examples means "in the current directory".

2) Assign passwords to the root accounts. Login as root:

```
> sudo ./mysql –u root
> mysql> SET PASSWORD FOR ''@'localhost' = PASSWORD('newpwd');
```

Where *newpwd* is the new password for the root account.

3) Assign passwords to the anonymous user accounts. Launch the mysql interpreter with this command:

```
./mysql -u root –password mysql
```

Here you are logging in as *root,* will be prompted for a password (the one you assigned above), and you'll be using the mysql database. You should see a display like this:

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8 to server version: 4.1.15-standard

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql>
```

Note: you can actually log into the server with any user name (or none) but you will have access to only a "test" database.

Note: All SQL commands are terminated with the semicolon character ';'. To continue a long command on a new line, simple press return—you'll be prompted for the rest of the command with an arrow "->". When you are finished composing a command type the semicolon character and press return.

4) Issue the GRANT command to create a new user:

```
mysql> GRANT ALL PRIVILEGES ON *.* TO 'oz'@'localhost'
    -> IDENTIFIED BY 'oz1234' WITH GRANT OPTION;
Query OK, 0 rows affected (0.35 sec)
```

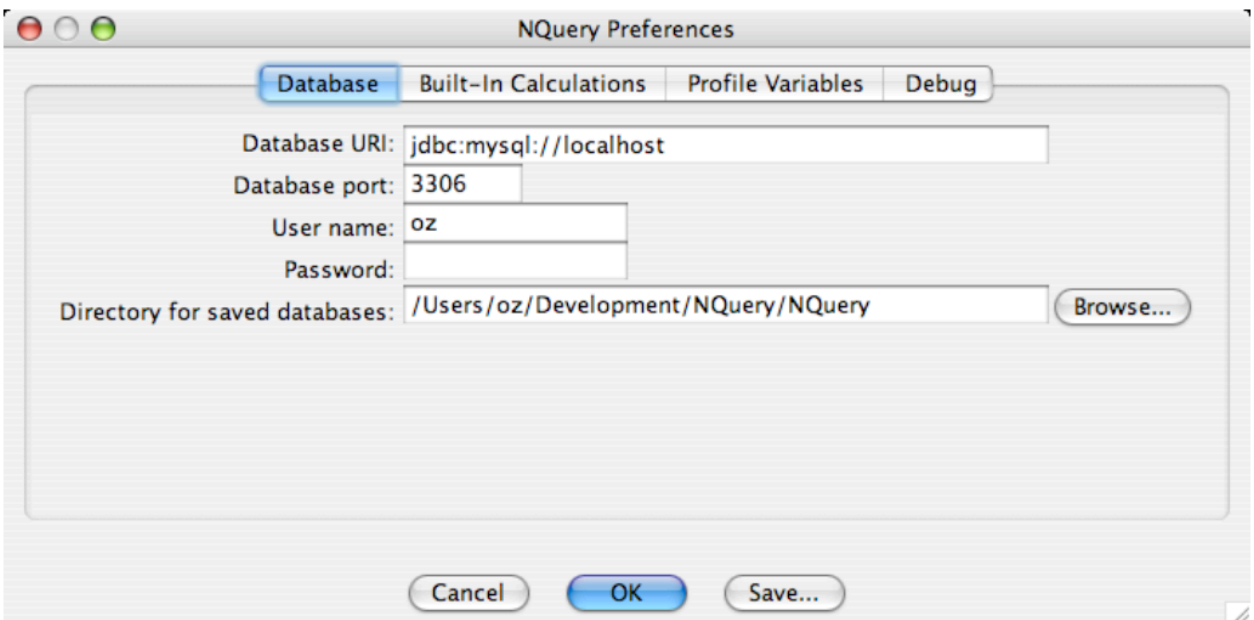
The above example illustrates creating a user named ‘oz’ with **all** privileges and password = ‘oz1234’. NQuery requires users to have all privileges.

Connecting to a Database Server

NQuery can connect to MySQL database servers running on your local machine or servers reachable via the Internet. Follow the instructions in Chapter 2 for instructions on installing MySQL on your desktop computer. The examples in this user guide assume that MySQL is installed on your desktop computer rather than connecting to a server over the Internet.

By default, NQuery tries to access a MySQL server at the URL: jdbc:mysql://localahost on port 3306. The URL for a distant server might look like: jdbc:mysql://dbserver.oceanatlas.com (a fictitious server). Port 3306 is the default for MySQL installations—you should check with the MySQL server administrator to verify the MySQL port number in use.

To change the MySQL settings, open the *NQuery Preferences* command from the *Edit* menu:

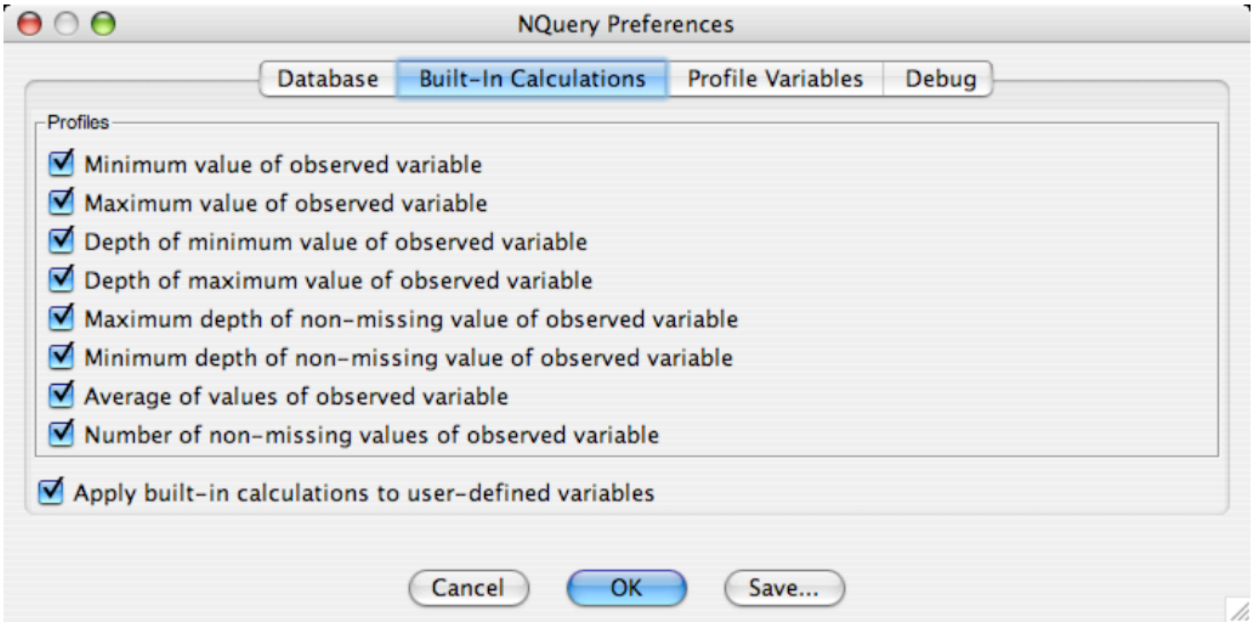


In *Database* panel you can change the URL of the database server, the port number of the server, assign a default user name on the database server, and set a default directory to store NQuery's saved databases (called database documents.) You can enter a password for the specified user name but it is only saved for the current NQuery session—it is not saved to the preferences file.

Clicking *OK* will allow any changes to persist for the current session. Click *Save*, to write your NQuery settings to the NQuery preferences file (with the exception of the password). Click *Browse* to set a different default location for saved database documents.

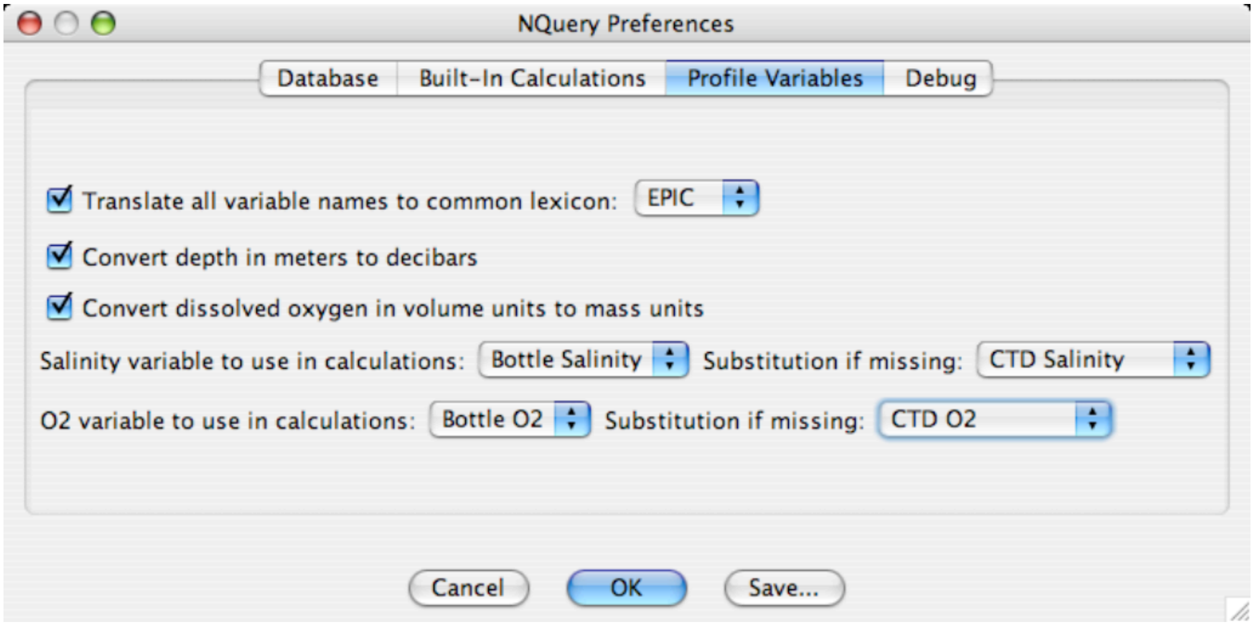
Setting Other NQuery Preferences

Specifying Built-In Calculations: NQuery includes 8 built-in summary calculations. By default all these calculations are selected. To change which built-in calculations are computed; choose *NQuery Preferences* from the *Edit* menu and click on the *Built-in Calculations* tab:



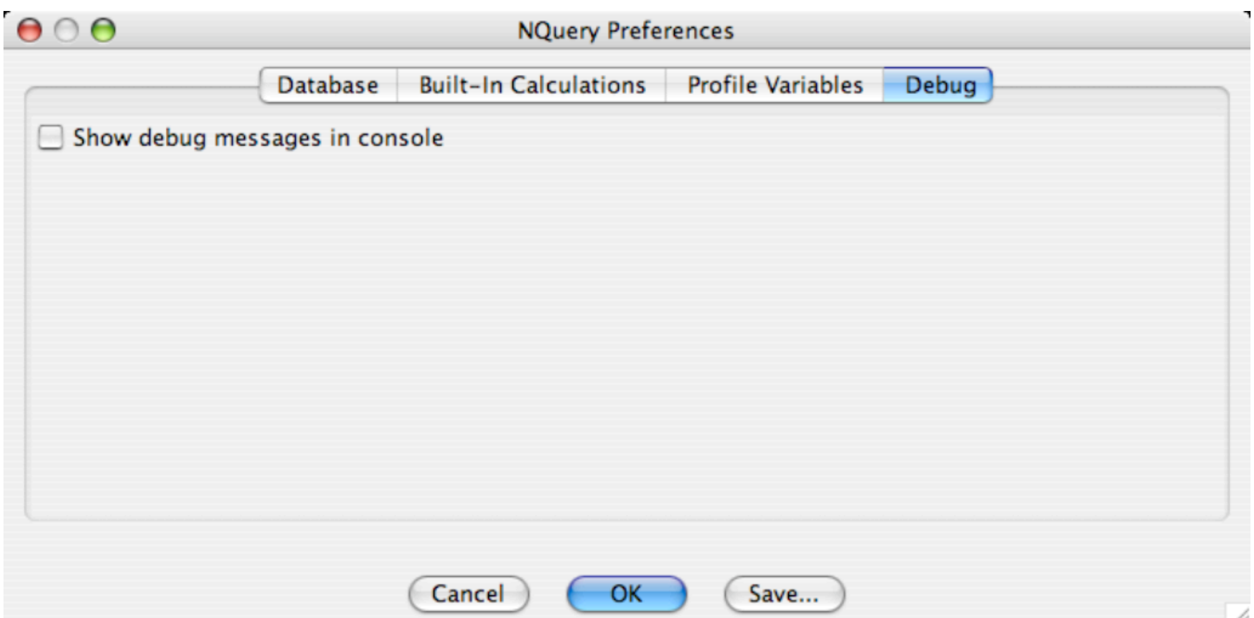
The selected calculations will be performed on all variables in the ingested data files plus any variables that NQuery computes from the observed data. Each calculation will result in a corresponding column in the database for each observed or calculated variable encountered. For example, if there are 5 measured variables in the ingested data, NQuery will build a table with 40 columns for the calculation results.

Profile Variables Options: NQuery is currently optimized for profile data. Because profile data can come from many different sources NQuery includes the capability to translate all variable names from pointer files and Dapper severs to a common lexicon. This allows you to display all variable names in one of three lexicons, EPIC, Java OceanAtlas (JOA), or WOCE. NQuery will recognize variables in Argo convention as well as deal with Dapper server conventions. In addition, NQuery allows you to define substitute bottle or CTD salinity, and dissolved oxygen for use in calculations. To set the profile variable naming convention select *NQuery Preferences* from the NQuery *Edit* menu and click the *Profile Variables* tab:



This panel also allows you to direct NQuery to convert the depth axis in meters to decibars and dissolved oxygen in volume units to mass units—units required by NQuery's algorithms for AOU, NO, and PO.

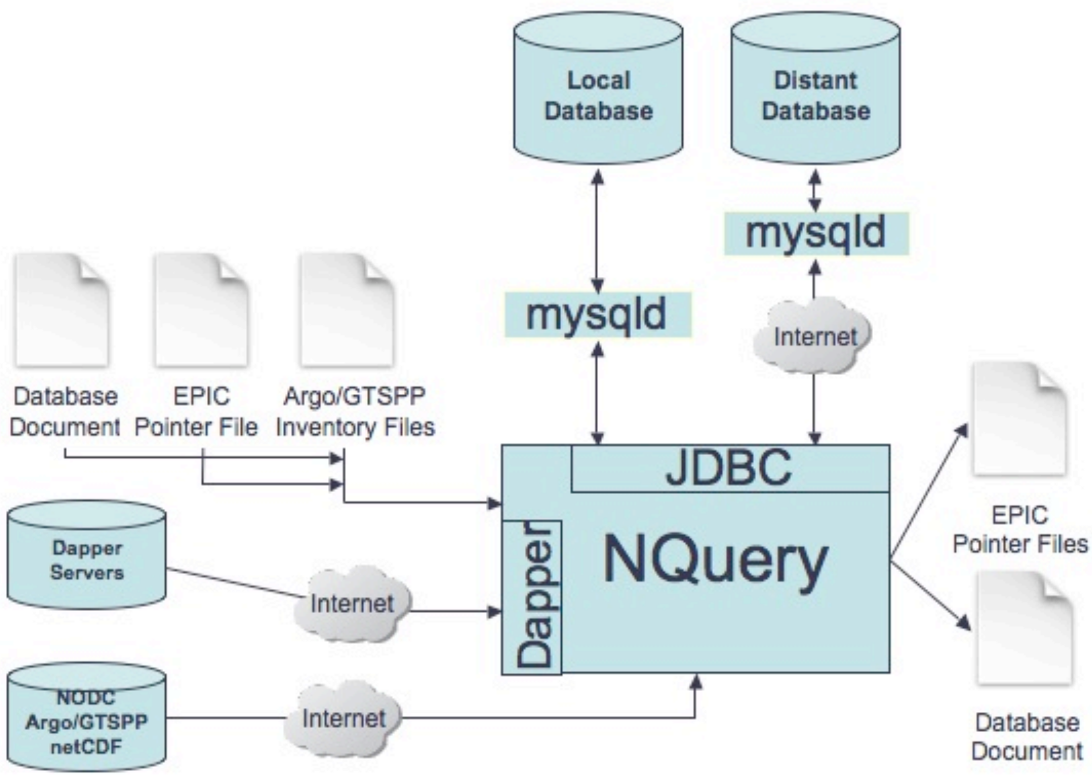
Turning on the Debug Output: If you are having trouble with NQuery, you can turn on NQuery's debug mode. To set the debug output mode, select *Preferences* from the NQuery *Edit* menu and click the *Debug* tab:



These messages will often help you figure out what's going wrong. In addition, the console output is useful to the developer when fixing bugs. Note: this setting is not saved to the preferences file.

Browsing Data in NQuery

Currently NQuery can access profile data from three primary sources: XML/EPIC pointer files, ARGO/GTSPP Inventory files, and *Dapper* data servers. The way data flows in and out of NQuery is summarized in this block diagram:



On the left of the NQuery symbol are data paths into NQuery. They include EPIC XML pointer files that typically describe data on your desktop PC, Argo/GTSPP inventory files that reside on your desktop but refer to actual data files on NODC servers, and Dapper servers that are accessed via the internet. In addition, *Database Documents* represent previous databases you have created with NQuery.

On the right are the output file formats; EPIC traditional and XML pointer files and Database documents.

The connections via the JDBC components are how NQuery connects with either local or distant MySQL servers.

EPIC XML Pointer Files: Pointer files typically contain spatial/temporal metadata and a field that “points” to an actual data file located on your desktop computer but can also point to files on the Internet. NQuery can read pointer files in the *EPIC XML* convention. These files define datasets of interests in the XML text markup language. Currently the only source of these files is Java OceanAtlas 4.x. Typically, you would use Java OceanAtlas to open data files on your desktop computer (or use online sources) to create a custom dataset that matches your research interest. You would then export a collection of individual netCDF files from JOA with an associated XML pointer file that describes the geospatial domain of the data as well as the location of individual data files. You can open this pointer file in NQuery to build a database from the profiles referred to in the pointer file. In addition to JOA, NQuery can produce EPIC XML pointer files from query results that point to files on your local computer or on the Internet. XML pointer files can be read into NQuery via the *Open XML Pointer File* command in the *File* menu.

Tip: You can compute many of NQuery’s calculated profile variables in JOA that are available in NQuery. Precomputing parameters like sigma, theta, etc… in JOA will reduce the time it takes NQuery to build a new database.

Here are the first few lines on an EPIC XML Pointer file:

```
<?xml version="1.0"?>
<epicxml version="1.0" type="profile" uri="file:///Users/oz/Desktop/JOA Test
files/WOCE Pacific Subset">
  <domain>
    <latitude location="south" units="degrees_north">-32.526</latitude>
    <latitude location="north" units="degrees_north">58.4987</latitude>
    <longitude location="west" units="degrees_east">124.9883</longitude>
    <longitude location="east" units="degrees_east">-
71.50170000000003</longitude>
    <vertical location="top" units="db" positive="down">0.0</vertical>
    <vertical location="bottom" units="db" positive="down">8996.0</vertical>
    <date location="start" year="1985" month="3" day="30" hour="5" min="42"
secs="0.0"/>
    <date location="end" year="1996" month="7" day="4" hour="19" min="9"
secs="0.0"/>
  </domain>
  <varlist>
    <variable name="PRES" units="db" lexicon="JOA" algorithm=""/>
    <variable name="TEMP" units="ipts-68" lexicon="JOA" algorithm=""/>
    <variable name="CTDS" units="pss-78" lexicon="JOA" algorithm=""/>
    <variable name="SALT" units="pss-78" lexicon="JOA" algorithm=""/>
    <variable name="O2  " units="ml/l" lexicon="JOA" algorithm=""/>
    <variable name="SIO3" units="um/l" lexicon="JOA" algorithm=""/>
    <variable name="NO3 " units="um/l" lexicon="JOA" algorithm=""/>
    <variable name="NO2 " units="um/l" lexicon="JOA" algorithm=""/>
    <variable name="PO4 " units="um/l" lexicon="JOA" algorithm=""/>
    <variable name="F11 " units="pmol/kg" lexicon="JOA" algorithm=""/>
    <variable name="F12 " units="pmol/kg" lexicon="JOA" algorithm=""/>
    <variable name="TRITUM" units="tu" lexicon="JOA" algorithm=""/>
    <variable name="C13 " units="/mille" lexicon="JOA" algorithm=""/>
    <variable name="O18O16" units="/mille" lexicon="JOA" algorithm=""/>
```

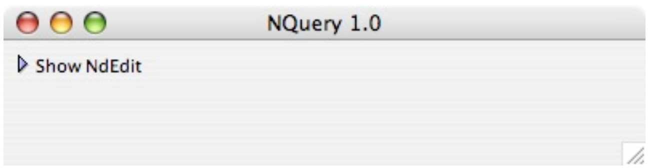
```
    <variable name="TCO2" units="umol/kg" lexicon="JOA" algorithm=""/>
    <variable name="ALKI" units="umol/kg" lexicon="JOA" algorithm=""/>
    <variable name="PH  " units="db" lexicon="JOA" algorithm=""/>
    <variable name="WTHT" units="deg" lexicon="JOA" algorithm=""/>
</varlist>
  <fileset id="P01W">
    <varlist>
      <variableref name="PRES"/>
      <variableref name="TEMP"/>
      <variableref name="CTDS"/>
      <variableref name="SALT"/>
      <variableref name="O2  "/>
      <variableref name="SIO3"/>
      <variableref name="NO3 "/>
      <variableref name="NO2 "/>
      <variableref name="PO4 "/>
      <variableref name="F11 "/>
      <variableref name="F12 "/>
      <variableref name="TRITUM"/>
      <variableref name="C13 "/>
      <variableref name="O18O16"/>
      <variableref name="TCO2"/>
      <variableref name="ALKI"/>
      <variableref name="PH  "/>
      <variableref name="WTHT"/>
    </varlist>
    <station id="30" cast="1" bottom="90.0" reference="P01W_30.nc">
      <date location="point" year="1993" month="9" day="12" hour="13"
min="11" secs="0.0"/>
      <latitude location="point" units="degrees_north">58.4987</latitude>
      <longitude location="point" units="degrees_east">141.8043</longitude>
      <vertical location="top" units="db"
positive="down">9.300000190734863</vertical>
      <vertical location="bottom" units="db" positive="down">49.5</vertical>
    </station>
    <station id="29" cast="1" bottom="148.0" reference="P01W_29.nc">
      <date location="point" year="1993" month="9" day="12" hour="9" min="32"
secs="0.0"/>
      <latitude location="point" units="degrees_north">57.9997</latitude>
      <longitude location="point" units="degrees_east">142.2978</longitude>
      <vertical location="top" units="db"
positive="down">10.199999809265137</vertical>
      <vertical location="bottom" units="db"
positive="down">74.69999694824219</vertical>
    </station>
    <station id="28" cast="1" bottom="190.0" reference="P01W_28.nc">
      <date location="point" year="1993" month="9" day="12" hour="5" min="37"
secs="0.0"/>
      <latitude location="point" units="degrees_north">57.4962</latitude>
      <longitude location="point" units="degrees_east">142.8107</longitude>
      <vertical location="top" units="db"
positive="down">10.600000381469727</vertical>
      <vertical location="bottom" units="db"
positive="down">99.30000305175781</vertical>
    </station>
```

Note: The output has been wrapped to fit the margins of this document. The specification for these files is summarized in Appendix B.

Example #1: Browsing an EPIC XML Pointer File

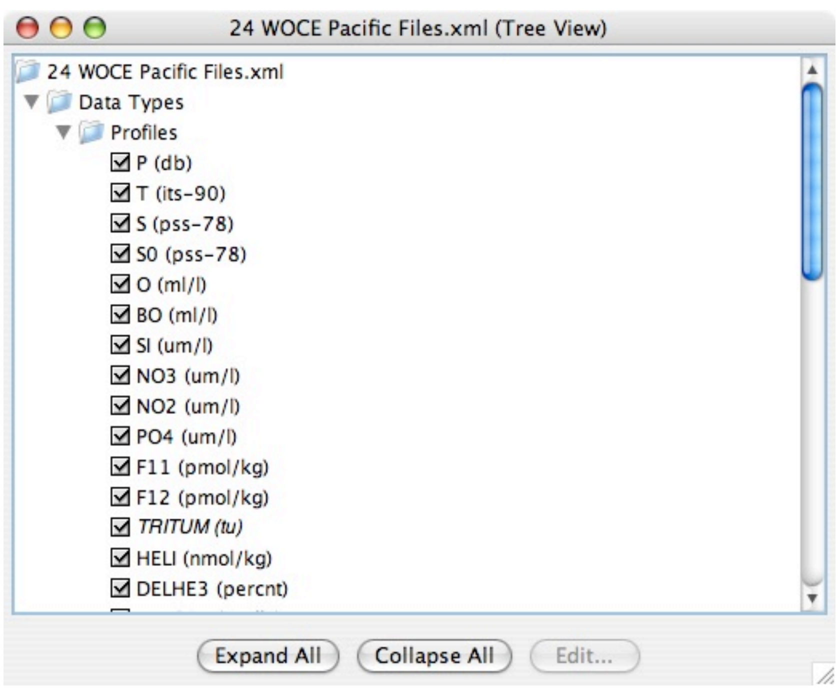
In this example, we will create a database by opening a pointer file that references data on your desktop computer. In this case, the data are all the WOCE profiles in the Pacific Ocean. To create this collection of 3,590 profiles, I opened all the individual section files in Java OceanAtlas and exported a folder of individual netCDF files tied together with a pointer file.

0) Launch the NQuery application. You will see the NQuery application window:

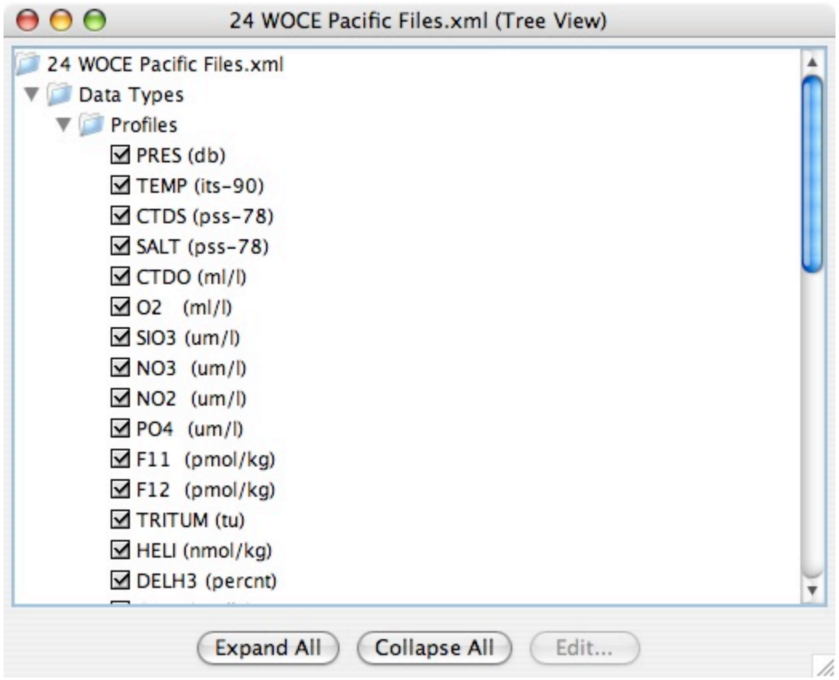


Note: This illustrates the Mac OS X look and feel; Windows and Linux will place the menubar in this window.

1) Open the XML pointer file in NQuery using the *Open XML Pointer File* command in the *File*). You will see the NQuery *Variable Inspector* window. Here you will select the variables to compute the built-in summary statistics on and add custom calculations:



Note: in the above illustration, variables names have been expressed in the EPIC lexicon. Notice that the variable TRITUM is rendered in italics. This is because that variable name does not exist in the EPIC variable lexicon so the original name is used. The next illustration shows what the variable inspector would look like if you didn't translate the variable names:



See later in this chapter for a complete discussion of NQuery calculations.

Example #2: Browsing an Argo/GTSPP Inventory Files

The latest inventory files can be downloaded from the Argo and GTSPP Web sites at:

http://www.nodc.noaa.gov/argo
http://www.nodc.noaa.gov/GTSPP

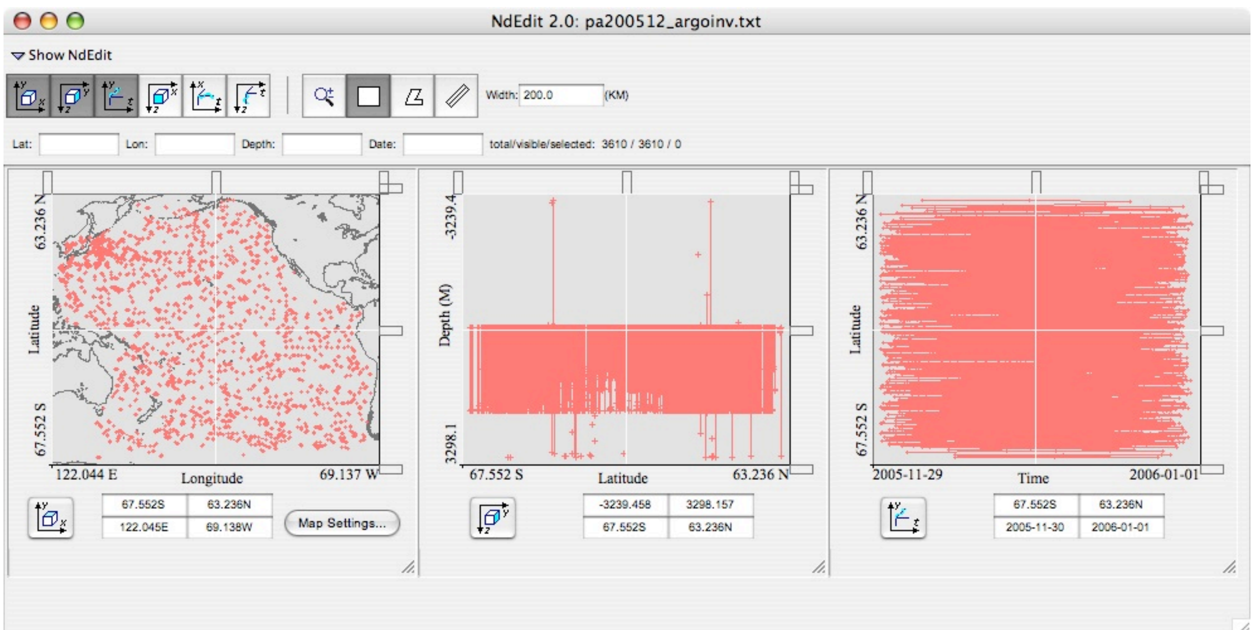
These files contain spatial/temporal metatdata and a URL to a netCDF file that contains the actual profile data. Here is an example of the first few lines of an Argo or GTSPP inventory file:

```
callSign,data_URL,file,ocean,date,time,time_qc,latitude,longitude,position_qc
,data_center,data_mode,num_of_levels,min_D_P,max_D_P,num_of_param,param1,para
m2,param3,param4,param5
1900148,http://www.nodc.noaa.gov/argo/data/pacific/2005/12/nodc_R1900148_104.
nc,pacific/2005/12/nodc_R1900148_104.nc,pacific,2005-12-04,17:23,1,-
53.363998,151.05499,1,AO,R,49,10.5000,949.000,3,PRES,TEMP,PSAL
1900148,http://www.nodc.noaa.gov/argo/data/pacific/2005/12/nodc_R1900148_105.
nc,pacific/2005/12/nodc_R1900148_105.nc,pacific,2005-12-15,06:22,1,-
53.508999,150.86099,1,AO,R,70,10.8000,1999.20,3,PRES,TEMP,PSAL
1900148,http://www.nodc.noaa.gov/argo/data/pacific/2005/12/nodc_R1900148_106.
nc,pacific/2005/12/nodc_R1900148_106.nc,pacific,2005-12-25,20:18,1,-
53.705002,151.01601,1,AO,R,50,11.0000,999.200,3,PRES,TEMP,PSAL
```

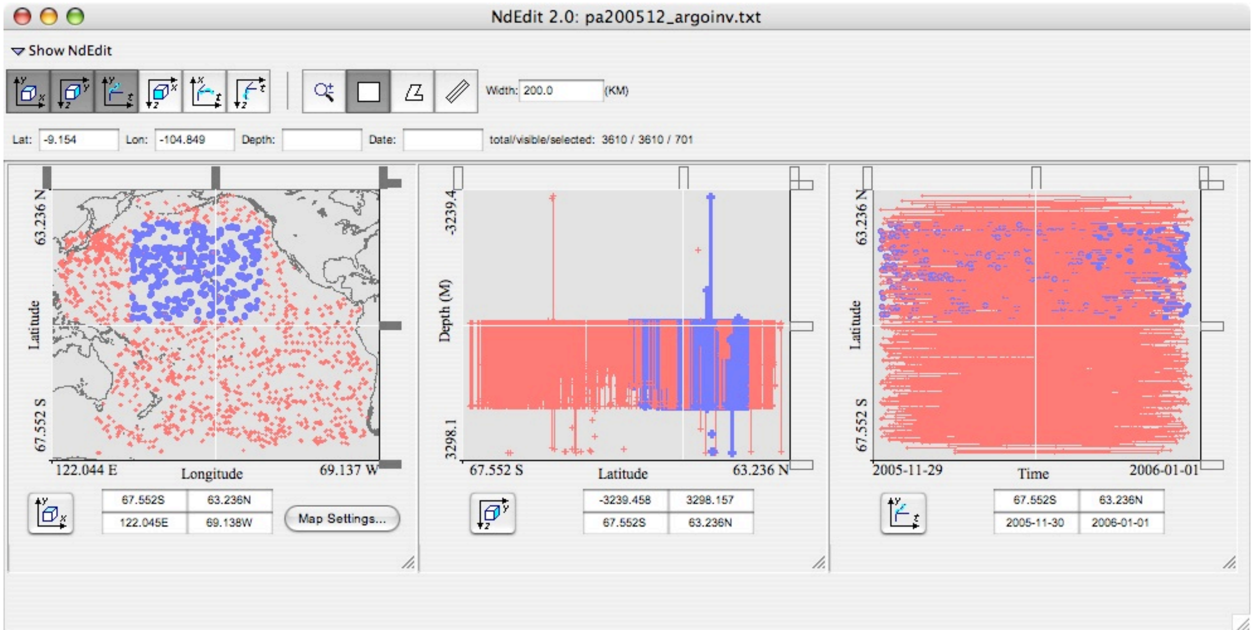
```
1900153,http://www.nodc.noaa.gov/argo/data/pacific/2005/12/nodc_R1900153_104.
nc,pacific/2005/12/nodc_R1900153_104.nc,pacific,2005-12-04,11:46,1,-
55.549000,161.98000,1,AO,R,50,10.7000,999.600,3,PRES,TEMP,PSAL
1900153,http://www.nodc.noaa.gov/argo/data/pacific/2005/12/nodc_R1900153_105.
nc,pacific/2005/12/nodc_R1900153_105.nc,pacific,2005-12-15,03:32,1,-
54.407001,163.89700,1,AO,R,69,11.0000,1949.00,3,PRES,TEMP,PSAL
1900153,http://www.nodc.noaa.gov/argo/data/pacific/2005/12/nodc_R1900153_106.
nc,pacific/2005/12/nodc_R1900153_106.nc,pacific,2005-12-25,13:44,1,-
56.813999,164.53799,1,AO,R,50,10.9000,998.900,3,PRES,TEMP,PSAL
2900139,http://www.nodc.noaa.gov/argo/data/pacific/2005/12/nodc_R2900139_117.
nc,pacific/2005/12/nodc_R2900139_117.nc,pacific,2005-12-
04,22:52,1,36.577000,153.28300,1,AO,R,72,3.80000,1398.80,3,PRES,TEMP,PSAL
```

Note: The lines of this file have been wrapped to the margins of this document.

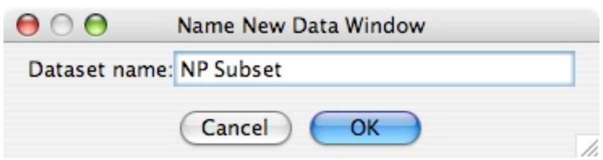
You use the *NdEdit* component of NQuery to filter the inventory files, select a subset of profiles, and ingest the selected profiles into NQuery. Inventory files can be read into NQuery via the *Browse Pointer File* command in the *File* menu:



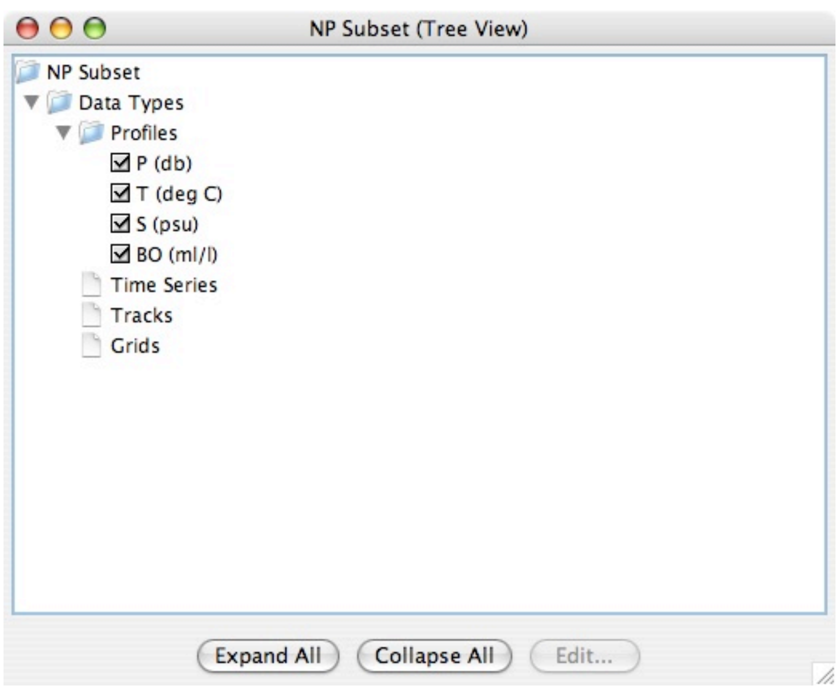
Select any number of profiles to create your database:



To open the variable inspector for this selection use the *Open Selection* command in the *File* menu. You will be prompted to name the new data selection:



You will see NQuery's Variable Inspector window:

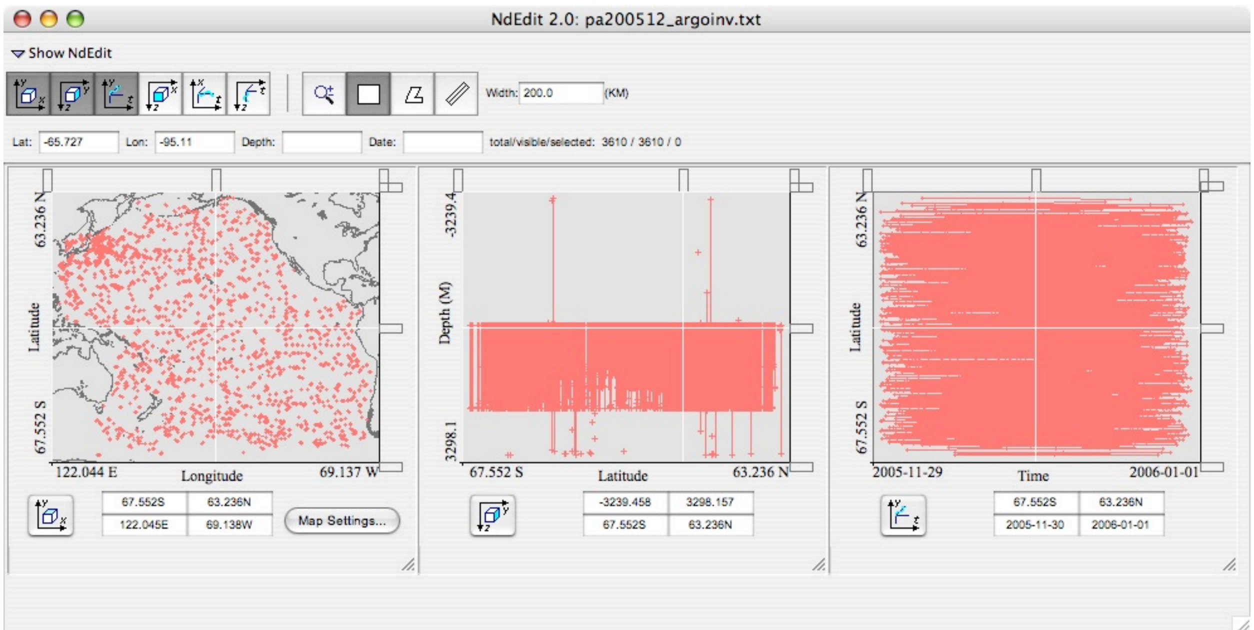


After you have selected a subset of Argo or GTSPP profiles, NQuery will use the URL in the inventory files to ingest the actual data file.

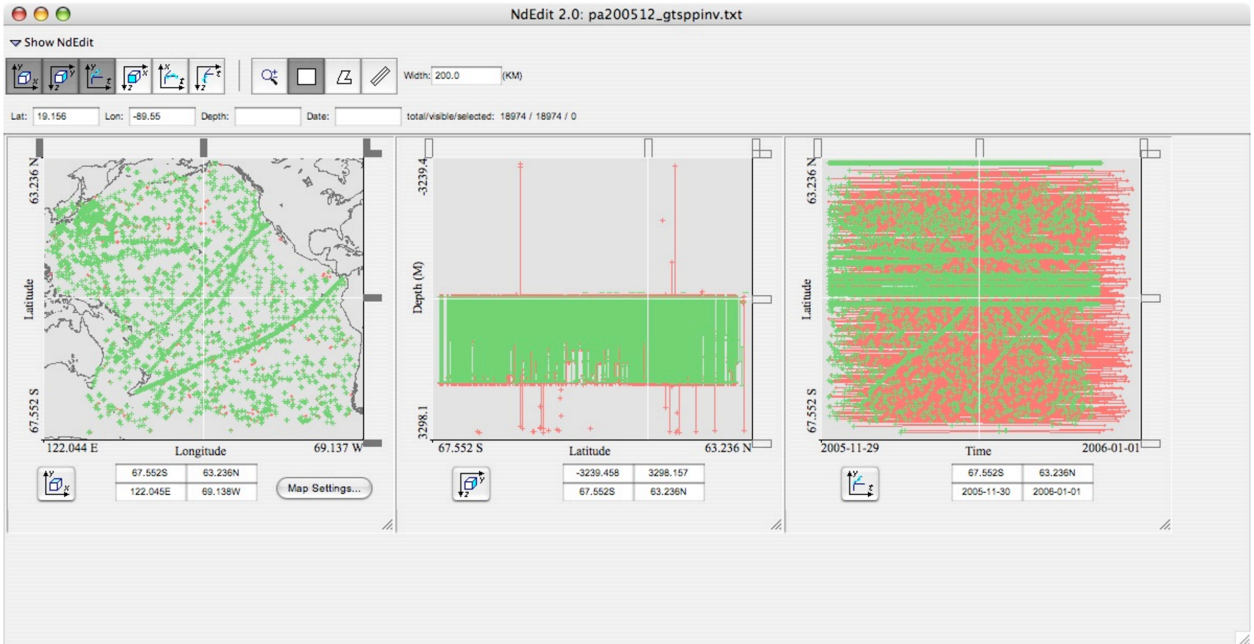
Example #3: Working With More Than One Inventory File

The NdEdit component can work with more than inventory file at a time. This allows comparison between different data sets and can make selecting synoptic datasets easier.

Open an Argo inventory file using the *Browse Pointer File* command in the menu:



Now use the same command and open a GTSPP pointer file:



The Argo profiles will be displayed in red while the GTSPP profiles will be displayed in green. The two pointer files are in separate virtual *layers* in NdEdit. You can now easily see how the datasets match up in time and space. To select data from NdEdit that has multiple data layers, select a data layer by right clicking (ctrl-clicking on Mac OS X without a two-button mouse) on any of the space-time panels. A popup menu appears that has the current layer checked:

Chose which layer you would like to select from by clicking its entry in the menu.

Limitation: NdEdit allows you to select from only one pointer file layer at a time and thus you are forced to create separate NQuery databases from multi-layer inventory files. A future version of NdEdit will allow selection from both inventories at the same time and thus will allow you to create an NQuery database that contains data from multiple sources.

Example #4: Working With Dapper Servers

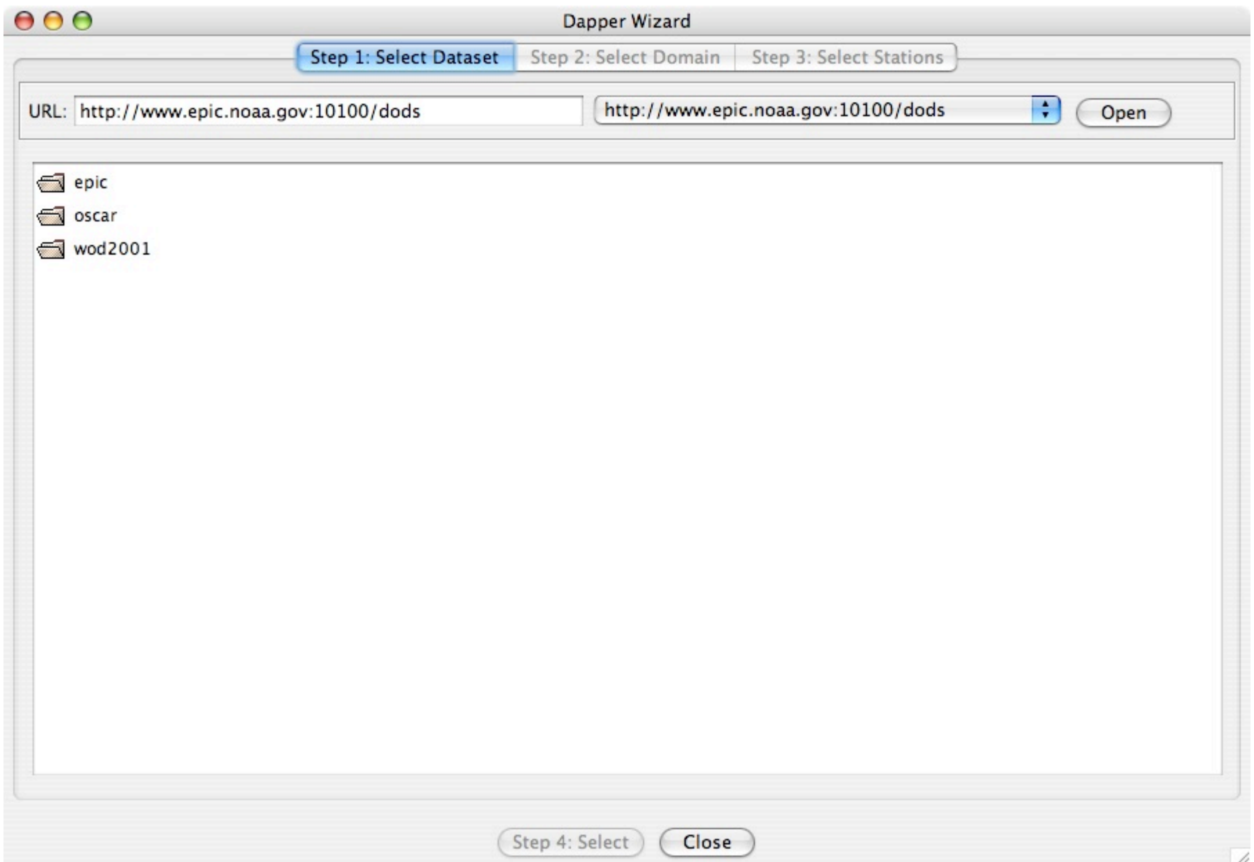
NQuery can browse and extract profile data directly from Dapper servers (http://www.epic.noaa.gov/epic/software/dapper/). Although, NQuery can ingest profiles, create on-the-fly databases, and query these databases, the query results are not yet directly useable in other applications because there are no applications that can read the Dapper URLs.

This example illustrates using NQuery to browse and extract data from a *Dapper* server. Dapper is a technology developed at NOAA/PMEL to allow access to large collections of profile data stored on central data servers. This example will illustrate extracting data from the NODC World Ocean Database 2001 on a server at NOAA/PMEL. Note: you must have access to the Internet to use the Dapper capabilities of NQuery.

1) Launch NQuery

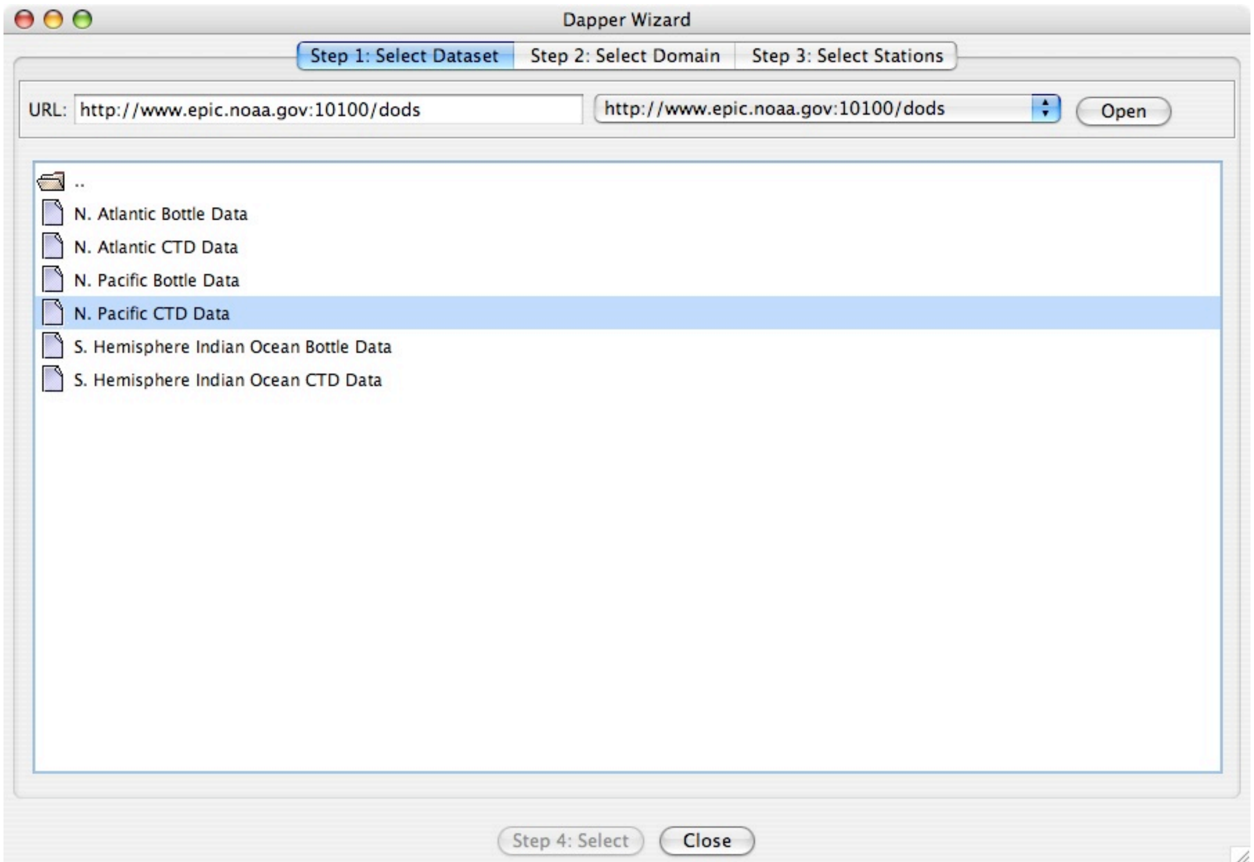
2) From the *File* menu select *Browse Dapper*. You will be presented with the *Dapper Wizard* that allows you to select a data collection, describe a geospatial/temporal domain in which to search for data, refine your selection using space and time filters, select profiles of interest, and open selected profiles in NQuery.

Here is the first step of the four-part Dapper Wizard:

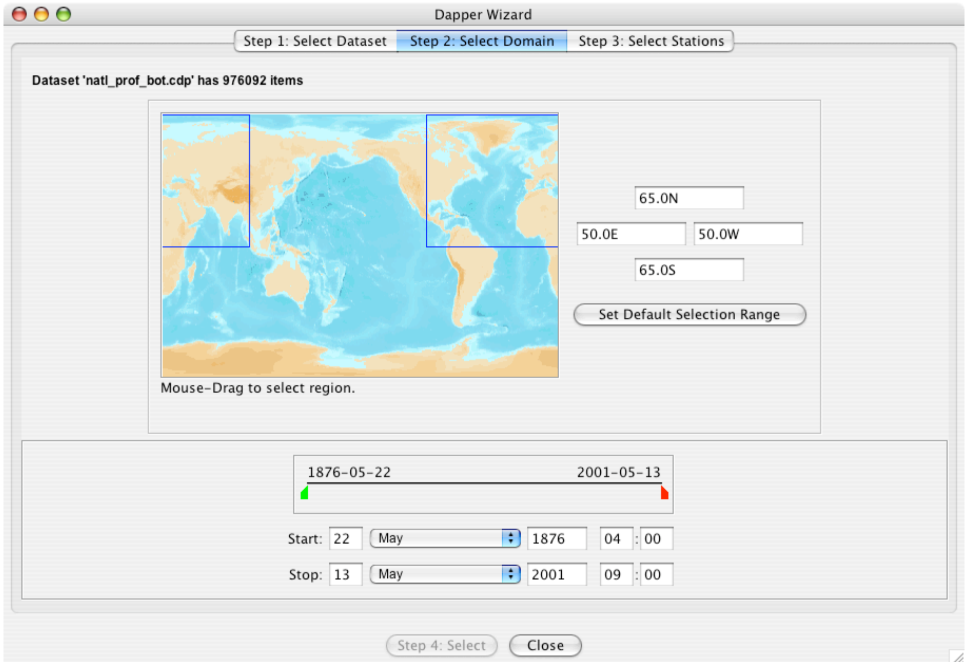


In this panel of the wizard, you enter the URL of a Dapper server (defaults to www.epic.noaa.gov:10100/dods) or select a different Dapper server from the popup menu. Click *Open* after selecting a Dapper server from the popup menu. Next select a data collection served at that address. In this example, there are three data collections available: *epic*, *oscar* and *wod2001*.

3) Double click the *wod2001* entry to browse datasets available in this collection. In the World Ocean Database 2001 collection, there are 6 datasets corresponding to ocean regions and data type–CTD or bottle profiles:

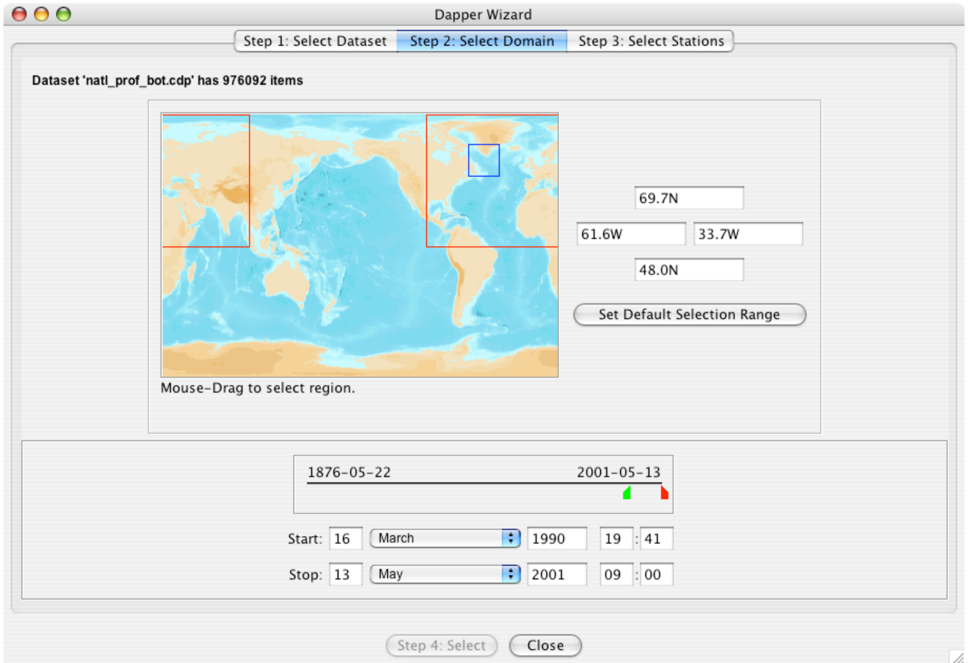


4) To go to the next step in the wizard, select a dataset and click the *Step 2: Select Domain* tab or simply double click the dataset of interest. To go back to the list of data collections, double click the top entry in the list (a folder icon with two dots). After a (sometimes lengthy) progress bar goes away, you will see the *domain selector* panel for the chosen dataset:

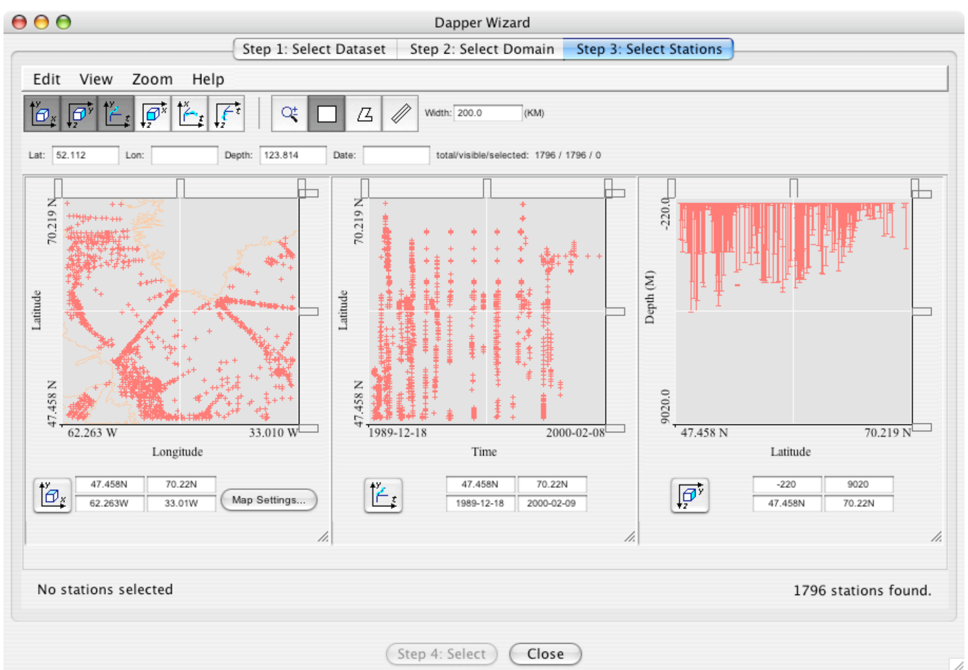


The map shows the spatial domain of the dataset and the time selector shows the temporal domain. Above the map is a label that lists how many profiles are in the dataset–in this dataset close to 1 million.

Because this is a large dataset, you will want to restrict the spatial and temporal domains. Dapper currently can return a maximum of 10,000 individual profiles. You can define a more restricted domain by dragging a rectangle onto the map (or entering explicit longitude/latitude coordinates in the fields to the right of the map) and using the handles on the time line control (or use the popup menus and text fields to enter an explicit date/time). For this example, select an area in the North Atlantic between Canada and Greenland and greatly reduce the time range:

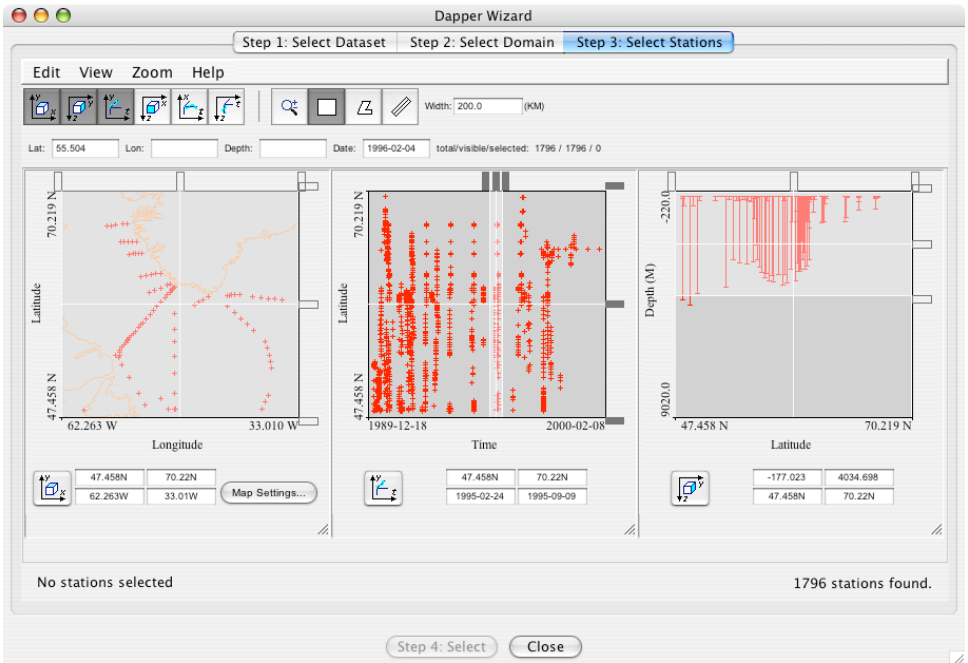


5) Click the *Step 3: Select Stations* tab to go to the fine selection panel of the wizard and select individual profiles for further analysis. At this point you will wait for the server to locate all the profiles in the space/time domain you specified. After the progress bar goes away you will see the individual profile locations displayed in the NdEdit data browser:

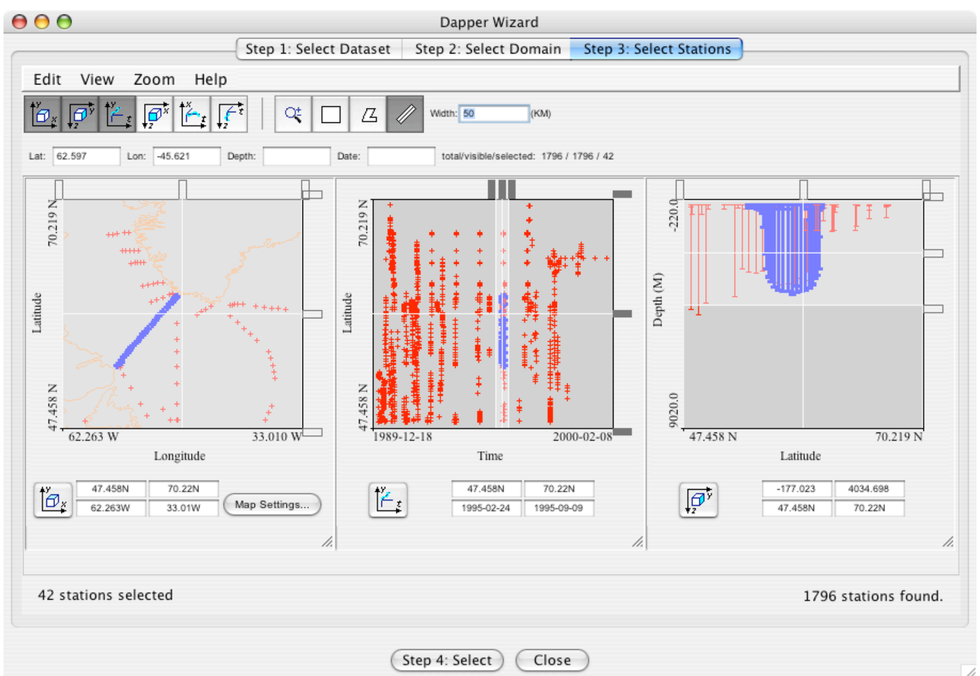


This search located 1,796 individual profiles. The default NdEdit view displays three panels for longitude/latitude, latitude/time, and latitude/depth slices of the dataset. If you have never used NdEdit, visit the NdEdit help and tutorials at http://www.epic.noaa.gov/epic/software/JavaNdedit.htm.

6) You can use NdEdit's interactive filters to reduce the number of visible profiles. Notice how many profiles have disappeared after narrowing the time domain:

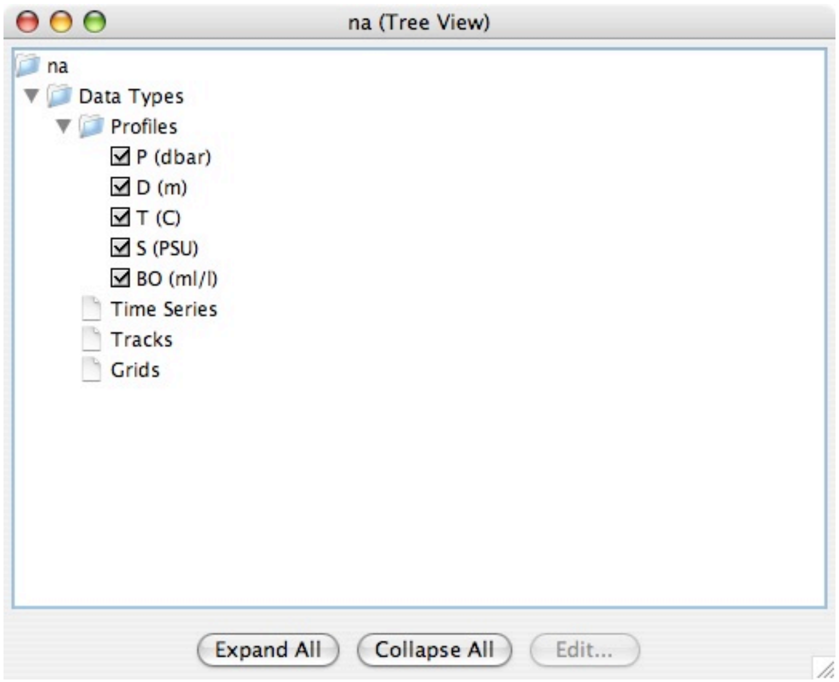


7 Use the rectangle tool to select all the displayed profiles in the Labrador Sea:



The 42 selected profiles are displayed in blue.

6) To open the selected profiles in NQuery, click the *Step 4: Select* button. The Dapper Wizard now will go back to the server and retrieve the metadata for each profile. NQuery will open a new Variable Inspector window:



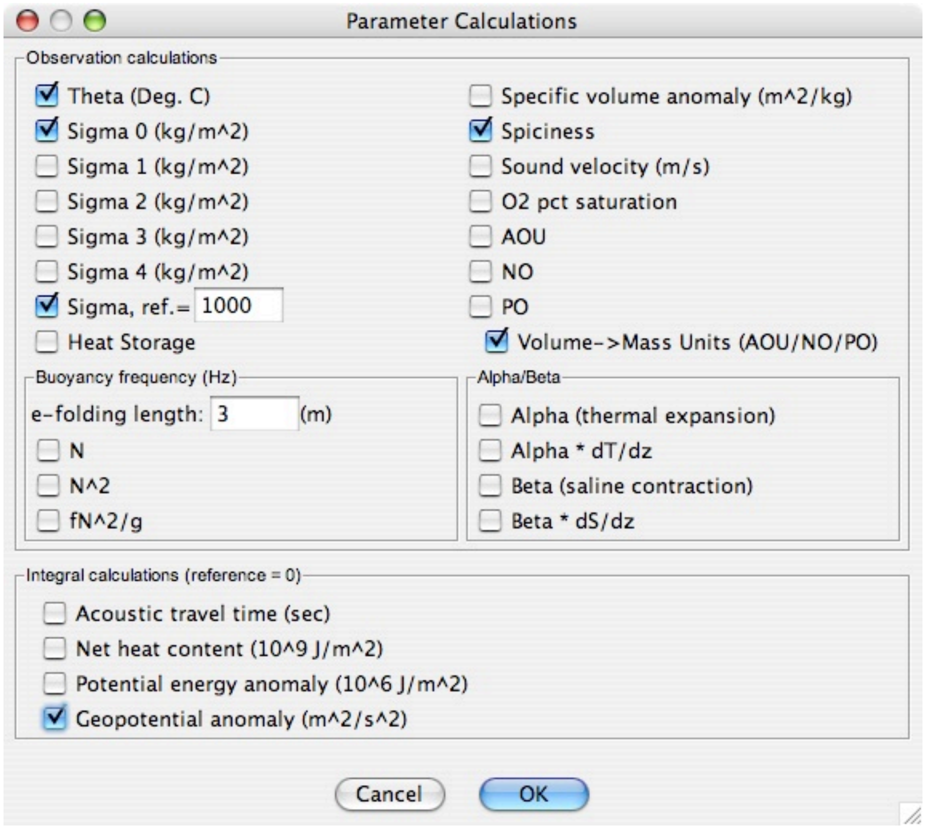
Calculating New Variables with NQuery

NQuery can calculate two type of new, user-defined, variables from the ingested data files; parameter calculations and "station" calculations. Parameter calculations result in a new value for each observation in the input data file—a new value at each observation or "bottle." Examples are theta, sigma-theta, AOU, NO, spiciness, buoyancy frequency, spiciness, and geopotential anomaly. After NQuery reads a profile, it calculates any observation calculations and then calculates the summary statistics for each observed variable and calculated variable.

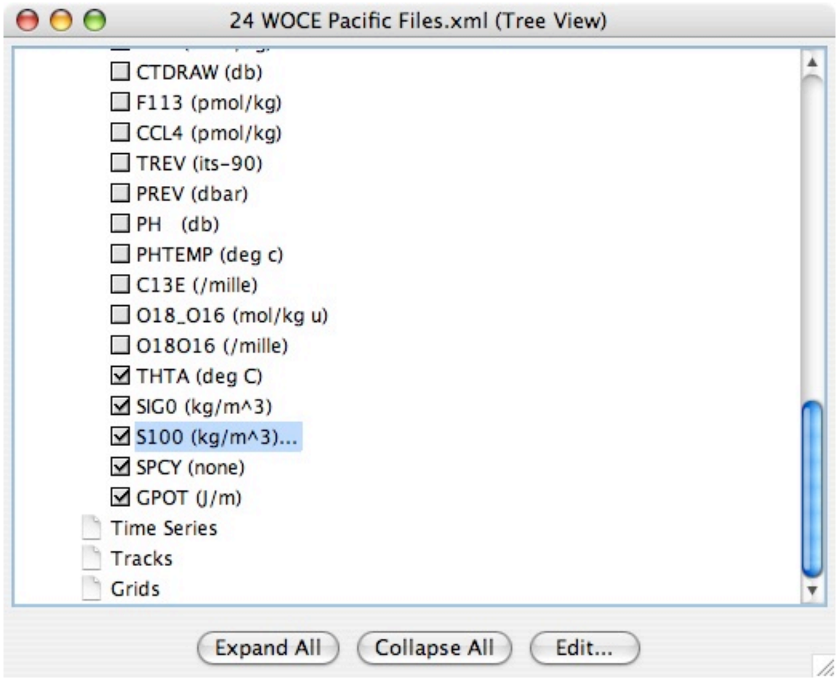
Station calculations represent a single value for each ingested profile. Examples include mixed layer depth, interpolation of an observed value to a surface of another observed variable (e.g., temperature interpolated to the surface), and integration of any observed variable in a range of another observed variable (e.g., average salinity in the range of 1000-2000 decibars). You can include as many station calculations as desired in the database.

Parameter Calculations

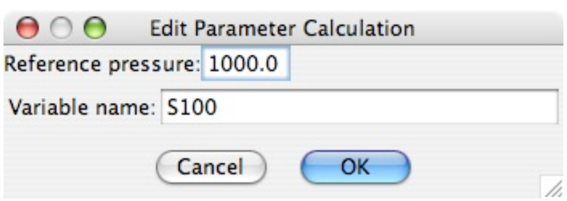
New parameter variables are added to the variable inspector with the *Parameter Calculations* command in the *Profile Calculations* menu. The following illustration shows which calculations are currently available:



New parameter calculations are added to the variable inspector window at the end of the variable list:



Calculations that have an argument (e.g., the reference pressure for the density calculation) can be edited by selecting them in the list and clicking the *Edit* button (can also just double-click the calculation in the list). In the above illustration, editing the S100 calculation presents the following dialog:

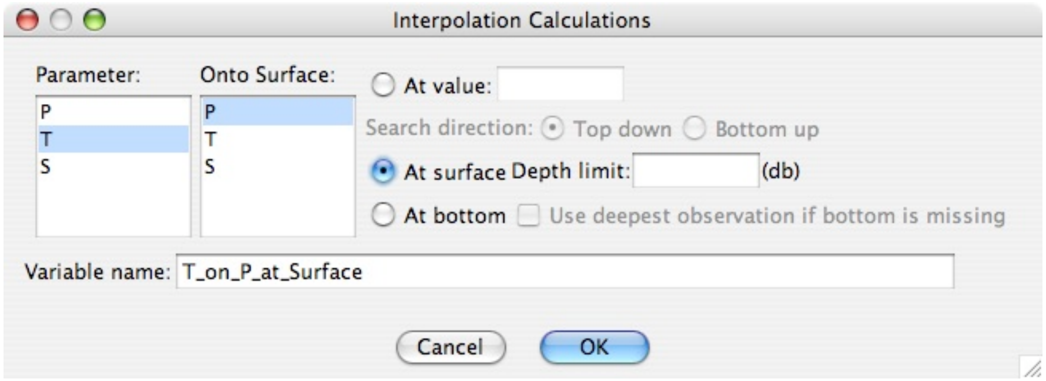


This dialog allows you to change the reference pressure and rename the calculation. Similar dialogs will be presented when editing other parameter calculations that require an argument.

Station Calculations

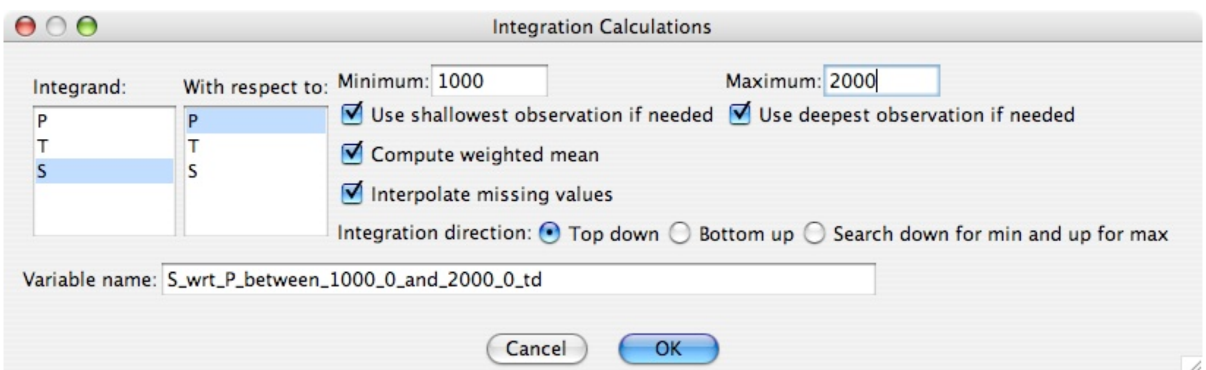
There are three types of station calculations, mixed-layer depth, interpolations, and integrations. These calculations are specified from the appropriate *Profile Calculations* submenu from the *Calculations* menu.

Interpolation: NQuery will interpolate one observed parameter to a selected value of any other parameter:



For example, it is possible to interpolate salinity to a value of pressure (depth) or density. It is also possible to compute the depth of an iso-surface of any other observed or calculated parameter by interpolating pressure onto the value of that parameter. For example, interpolating pressure to a value of temperature equal to 10.0 will return the depth of the 10-degree isotherm. If you are interpolating to pressure, NQuery has some additional options that allow you to interpolate to the surface or the bottom of the ocean. Interpolations to the surface have the option of defining a depth limit so that the first observation of a deep cast is not considered the surface. When interpolating to the bottom, you can specify whether to use the deepest observation if the depth datum for that station is missing.

Integration: NQuery will calculate the integral (step-wise) of one original or calculated parameter over a selected range of any other parameter:



Options are presented to

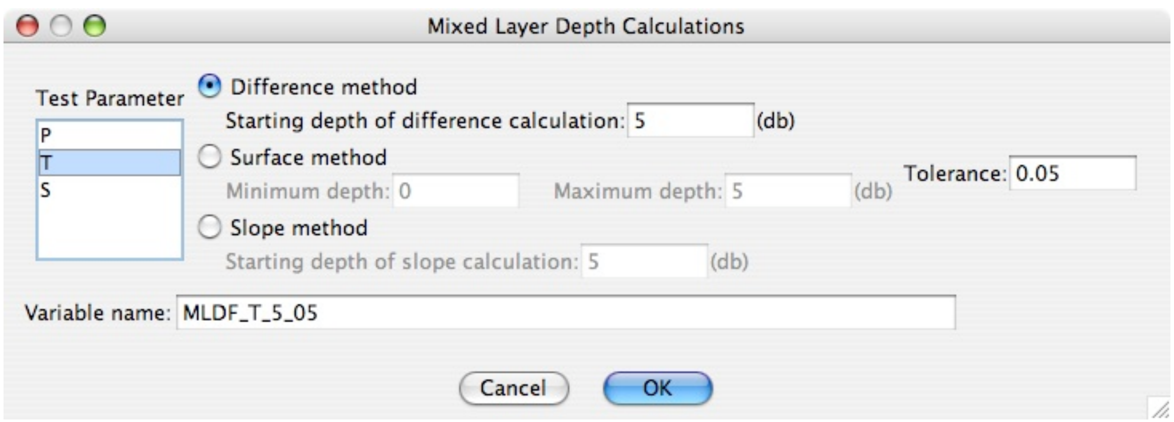
> *Use the shallowest (or deepest) observation if needed* - for example if integrating over a sigma-0 range and the density at the shallowest observation is greater than the starting value should the values at the shallowest observation be substituted or should the calculation result be reported as missing for that station?
>
> *Compute weighted mean*: weight the result over the limits of integration (otherwise the results are left raw). This results in an integration result in the units of the original integrand.

Interpolate missing values: if an observation of the integrand parameter is missing inside the range of integration, should NQuery provide a linear interpolation at the level from the closest levels above and below with non-missing observations? Can also set a custom value of *Max number of observations* to limit interpolation of missing values.

The integration direction can be specified to be from shallow to deep, from deep to shallow, or by searching down for minima of the parameter of the axis of integration and up for maxima of the parameter of the axis of integration.

Warning: The integration calculation is not sophisticated. It works reasonably well for ranges of any parameter, which does not contain mid-profile extrema (it works well over pressure or density, for example). Despite the logic choices, the results can be misleading when a parameter is integrated over an axis with mid-level extrema. The intent is to supply for a typical data file a reasonable approximation of integration calculations.

Mixed-Layer Depth: NQuery will calculate the depth of the mixed layer for each ingested profile and add to the database:



Any original parameter can be used as the test parameter. The *Test Parameter* is the parameter whose variability - or lack of it - is used as the indicator of mixed-layer depth.

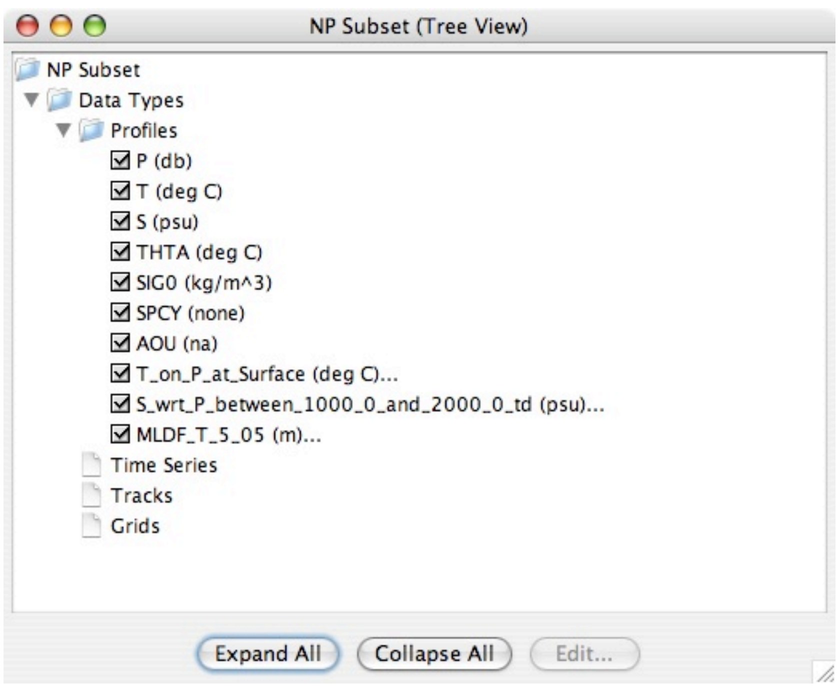
Three methods of mixed-layer depth calculation are available:

In the *Difference method* NQuery, beginning at the *Starting depth of difference calculation*, will look at successively deeper (higher pressure) observations in each data file until it reaches an observation whose value for the test parameter differs from the value of the test parameter at the starting depth by more than a user-chosen *Tolerance*. For example, if 'T' is selected with a starting depth of 5 and a tolerance of 0.5, NQuery will report back the thickness between 5 and whatever level contains a temperature 0.5 degrees different than the value at 5.

The *Surface method* is similar to the difference method except that instead of comparing the data down the water column to the value at the starting depth, NQuery compares the data to a calculated bulk surface value that is created by averaging the test parameter over the range of pressures specified. The depth of the mixed layer is reached when the difference between a observations's value of the test parameter and the bulk surface mean is greater than the tolerance.

The *Slope method* is similar in some respects to the difference method, but each successive measurement pair is used for the examination of whether or not their difference exceeds the tolerance. For example, if 'T' is selected with a starting depth of 10 and a tolerance of 0.05, NQuery will compare the next deeper measurement to 10, the one after it to the one after 10, etc., and report back the thickness between 10 and whatever level contains a temperature 0.05 degrees different than the previous value.

Warning: These algorithms are best suited to CTD profiles. The intent is to supply for a typical data file a reasonable approximation of mixed layer calculations.
Editing Station Calculations: After configuring a station calculation, it is added to the variable inspector at the end of the variable list:



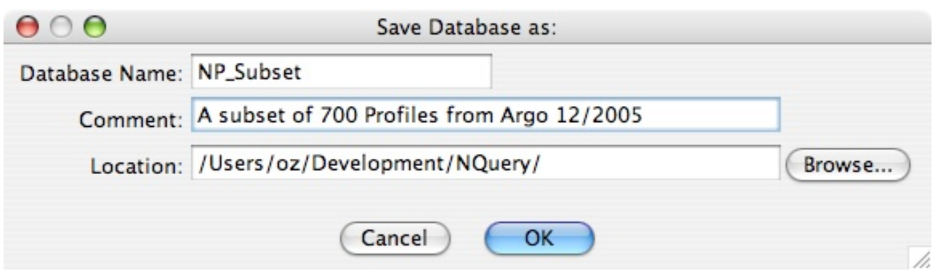
To edit an existing station calculation, select the calculation in the list and click the *Edit* button (or double click the calculation.) The appropriate dialog will open that will allow you to change the settings of the calculation as well as it's name.

Currently, there is no way to remove a station or parameter calculation from the variable list. If you change you mind about whether a calculation is needed simply uncheck the box next to it in the list.

Creating a New Database

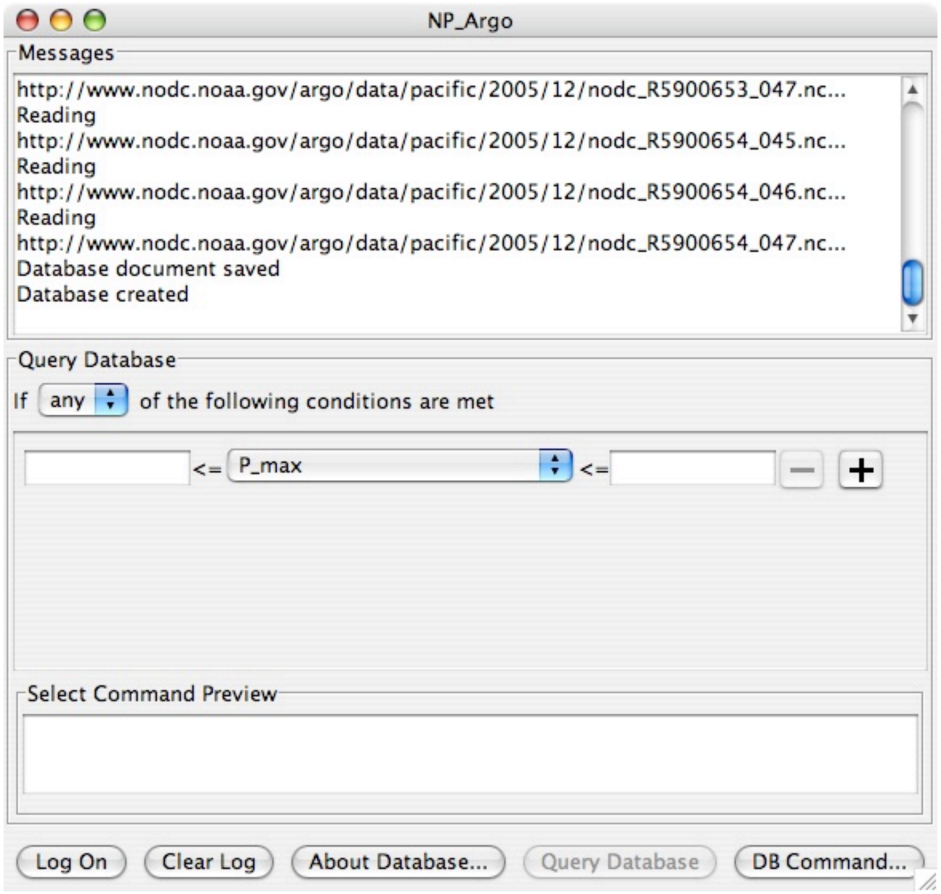
After you have selected variables and defined custom calculations, you are ready to create a new MySQL database from the selected profiles. With the Variable Inspector window selected, issue the *Create Database* command in the *File* menu. If you haven't entered a password yet, you will be prompted to do so:

You will next be prompted to name your new database and provide an optional comment. The database name will be used to name the actual MySQL database. The database name and comment will also be stored in the Database Document created with every database so existing databases can be reopened. You can chose to save the Database Document in the specified location or chose a new location by clicking the *Browse* button.



Note: the database name should only contain standard upper and lowercase letters and numbers. Special characters should be limited to the underscore and dash characters. Using other punctuation characters and/or spaces will cause an error creating the database.

Click *OK* to build the database structure, ingest the selected profiles, calculate summary statistics, calculate user-defined variables, and populate the database tables. A new Database Document will open and the progress of data ingest will appear in the *Messages* are of the window. If there are no errors, you will see *Database document saved* and *Database created* in the message console:



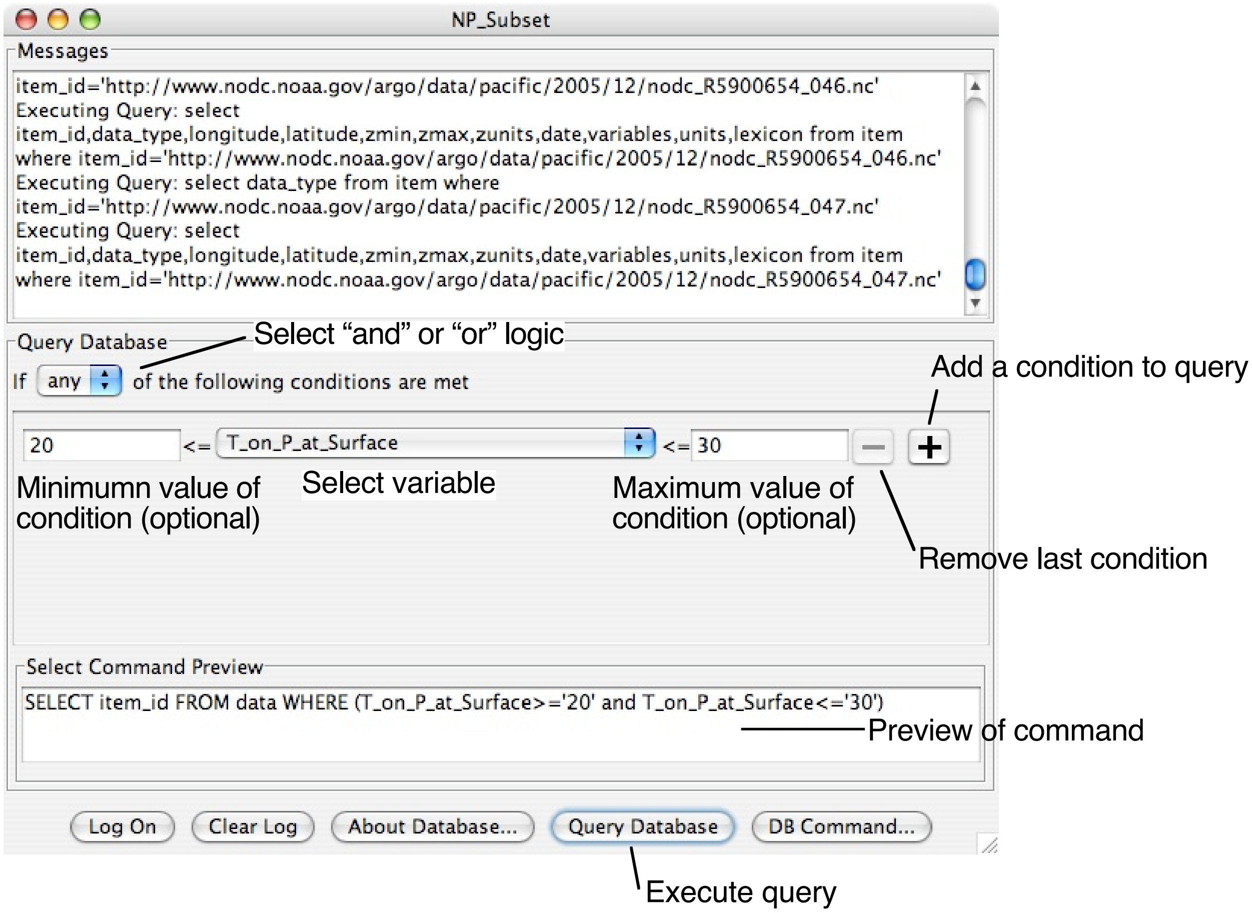
Opening Existing Databases

When NQuery builds an on-the-fly database, it also creates a separate document that allows you to reopen the database. This is called a “database docment”. These files have an .nqdb extension and are stored in the directory specified in the NQuery preferences dialog. After a database document has been created, it can be reopened via the *Open Existing Database* menu in the NQuery *File* menu.

Searching NQuery Databases

The primary purpose of NQuery is to identify data of interest by its characteristics using the summary statistics computed from the observed values and any user-defined calculations such as mixed layer depth. NQuery provides two ways to query a database; graphical and command line.

The graphical user interface for building queries is located in the middle panel of the database document window. The labels in the following figure illustrate the features of the query builder:



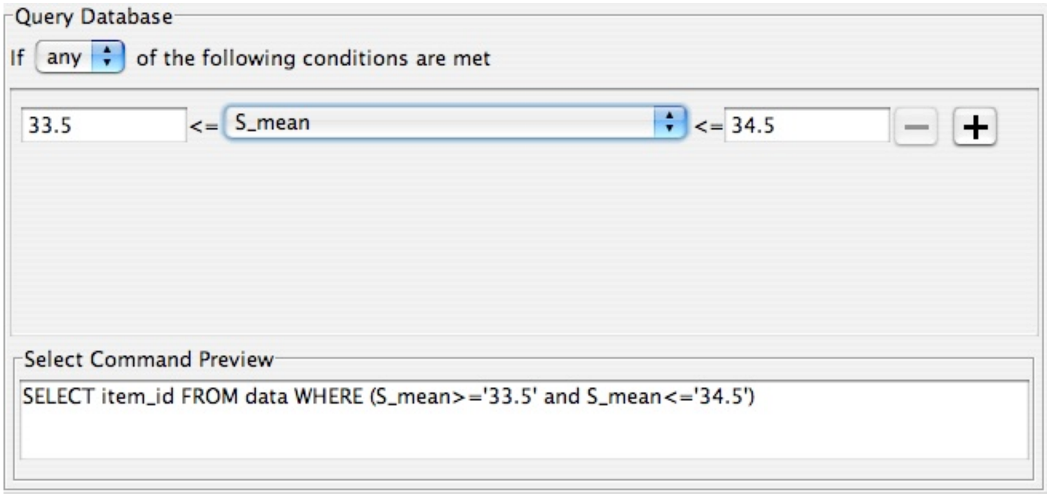
Search logic: A search can be constructed with either "or" or "and" logic. Selecting "any" from the popup menu means that individual conditions will be connected with "or" logic and will broaden the results of the query. Selecting "all" from the popup menu means that individual conditions will be connected with an "and" statement and will narrow the results of the query.

Query Builder: Each condition is composed of a variable from the database (actually a column in a table) and a range of values to test that variable's values against. Select a variable from the center popup menu and provide at least a minimum value or a maximum value or both. To add a new condition, click the large "plus" button. To remove the last condition, click the large "minus" button. Note you can not remove the first condition.

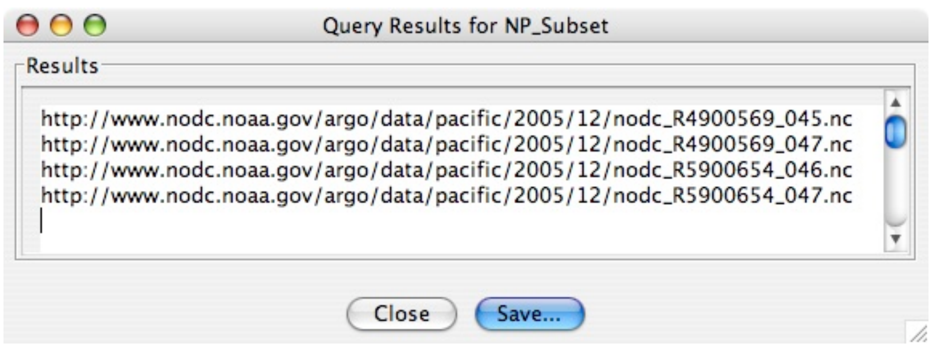
Query Preview: The exact SQL query is constructed as you use the query builder and is shown in the Select Command *Preview* area of the window.

A query can have as many conditions as you want but note that the graphical query builder can not group conditions with parentheses or include a "not" operator. If you wish to construct a query out of the scope of the graphical query builder, click the *DB Command* button to enter your query expression.

Example of a Simple Query: In this example we are looking for all the profiles in a certain range of salinities:

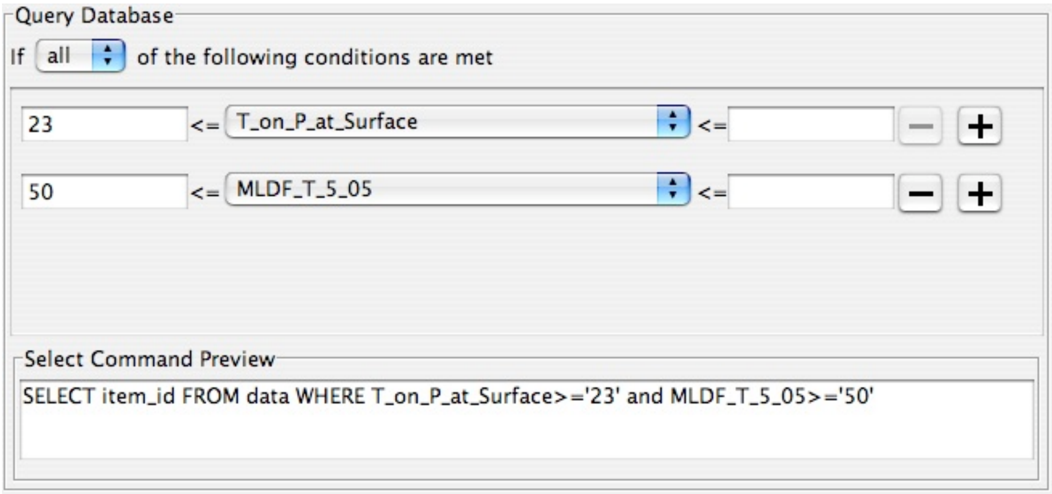


Which returns the following results:

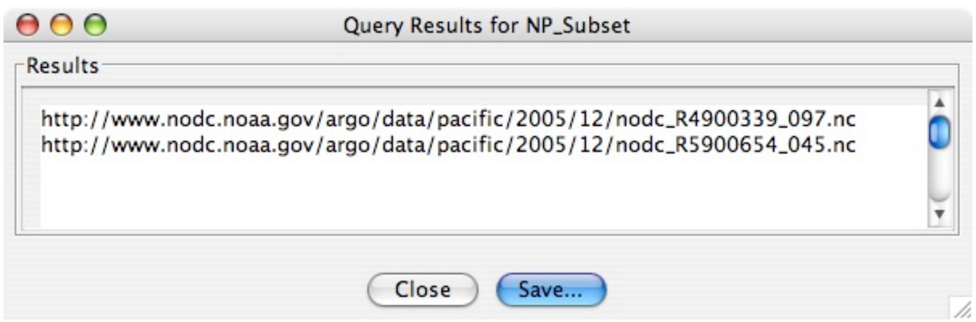


Clicking *Save* would allow you to save the results as an EPIC pointer file and open this set of files in Java OceanAtlas.

Example of a Complex Query: A complex query allows you to test the values in the database against two or more conditions. This example tests whether the interpolated sea-surface temperature is greater than 23° AND the computed mixed layer depth is greater than 50m:



Notice that the *all* option is selected in the logic popup to use “and” logic in the query. This query finds only 2 matching profiles:

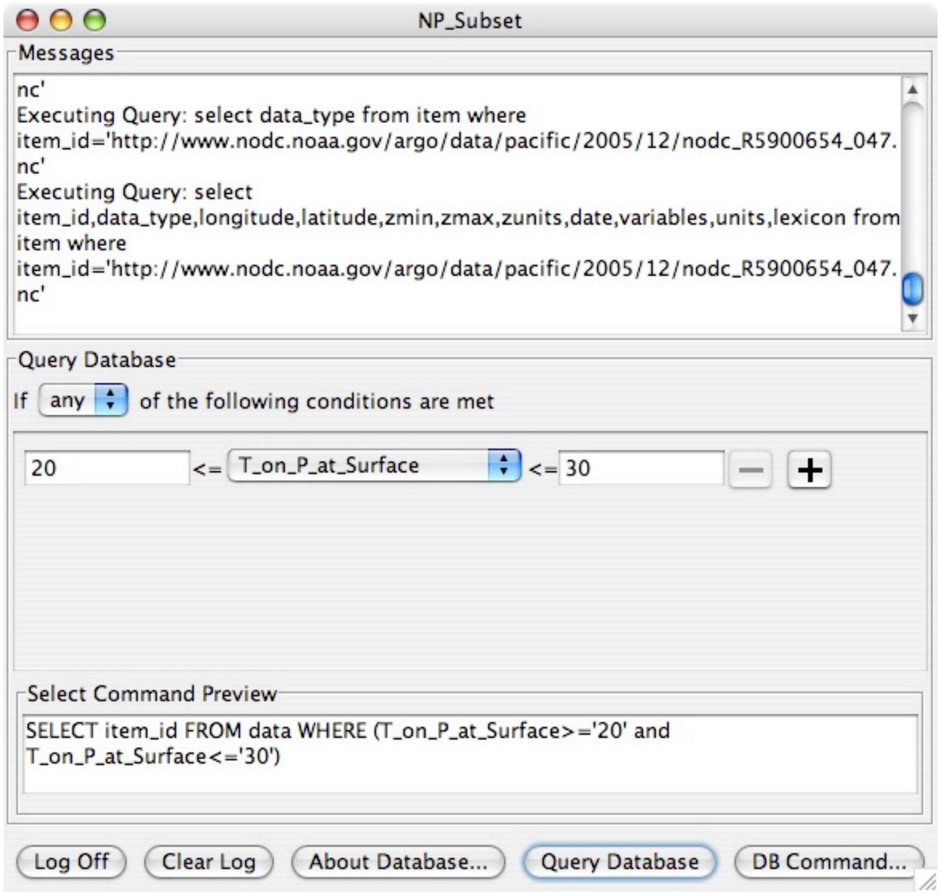


Saving and Exporting Query Results

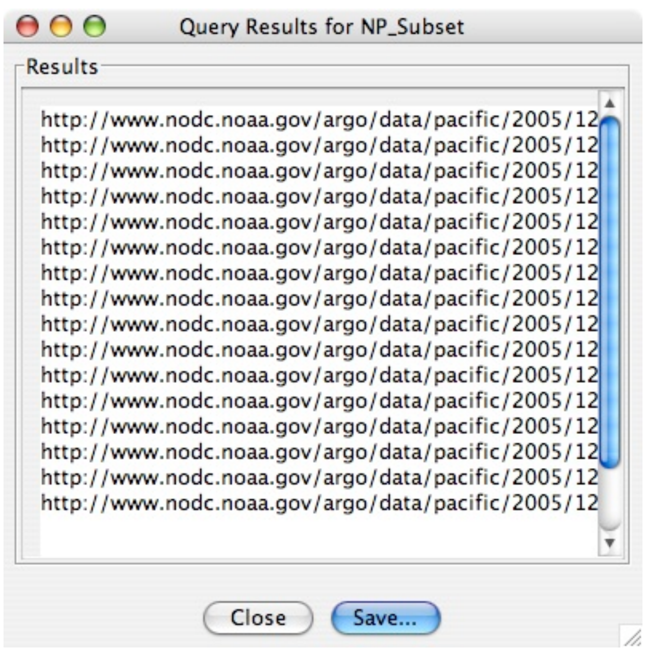
Query results can be written to either a traditional EPIC pointer file or a new XML pointer file. Currently, only Java OceanAtlas 4.1 can directly read these files and only for profile files that can be located on your local machine or files reachable with an HTTP URL like the Argo and GTSPP archives at NODC.

Example: Saving query results and opening results in Java OceanAtlas

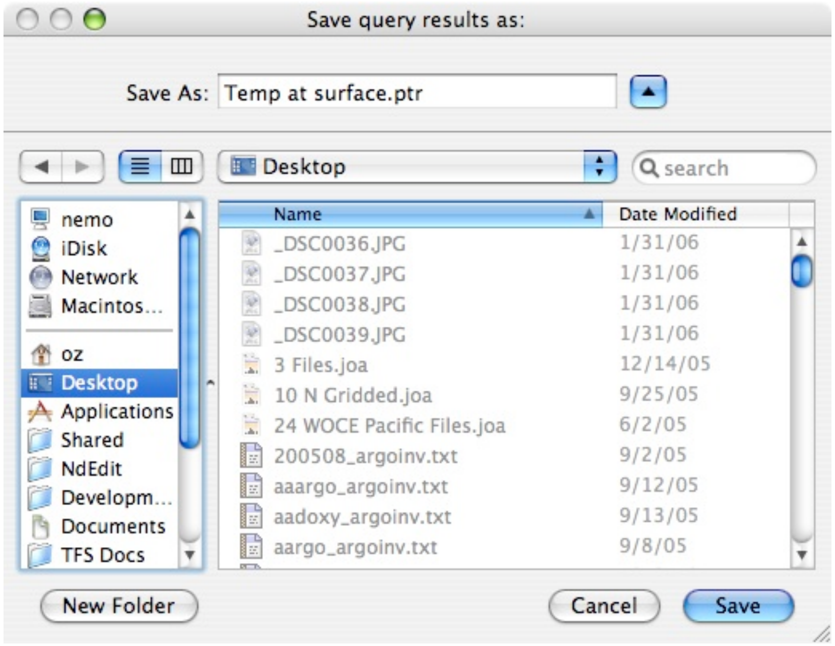
Consider the following query:



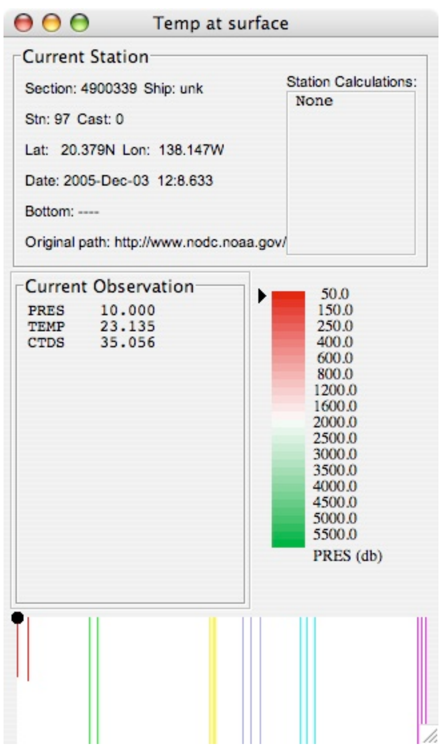
Produces these results:

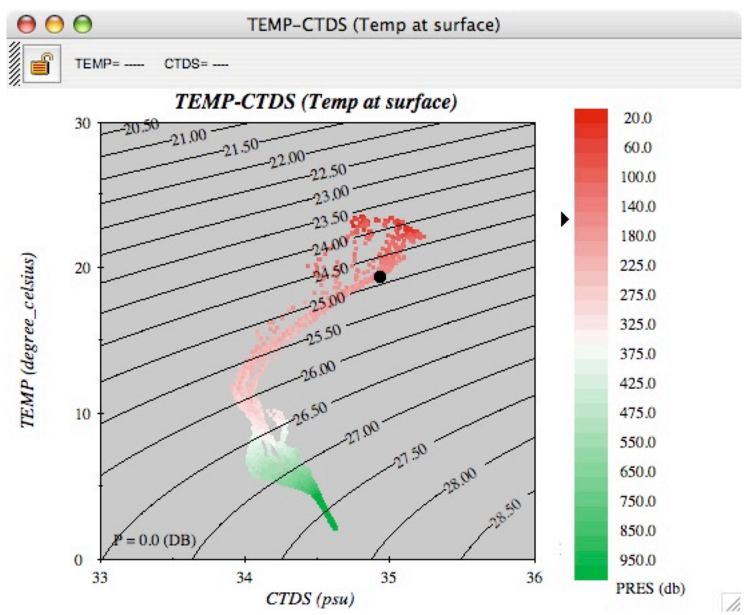


Click the *Save* button and specify a file with a file with a .ptr extension:



Open the pointer file in Java OceanAtlas 4.1 and produce a T-S plot:

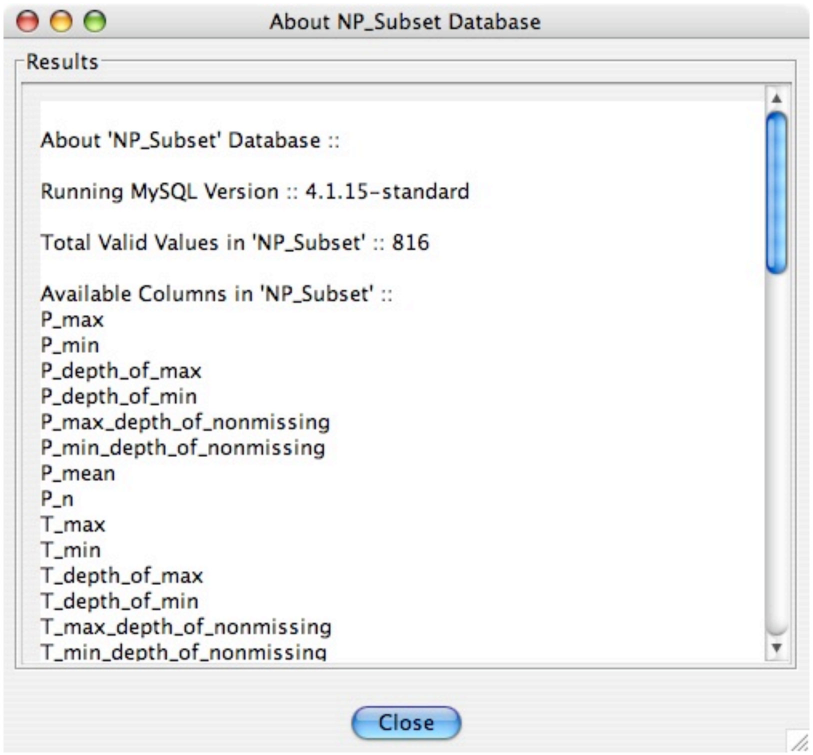




CHAPTER 5. MAINTAINING NQUERY DATABASES

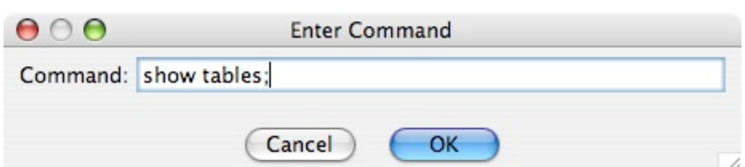
Getting information about the Current Database

To find out general information about the current database (the one open in the active database document window), click the About Database button in the database document window. A window with a scrolling list of the total number of values and column names in the current database will be opened:



Issuing MySQL Commands From Within NQuery

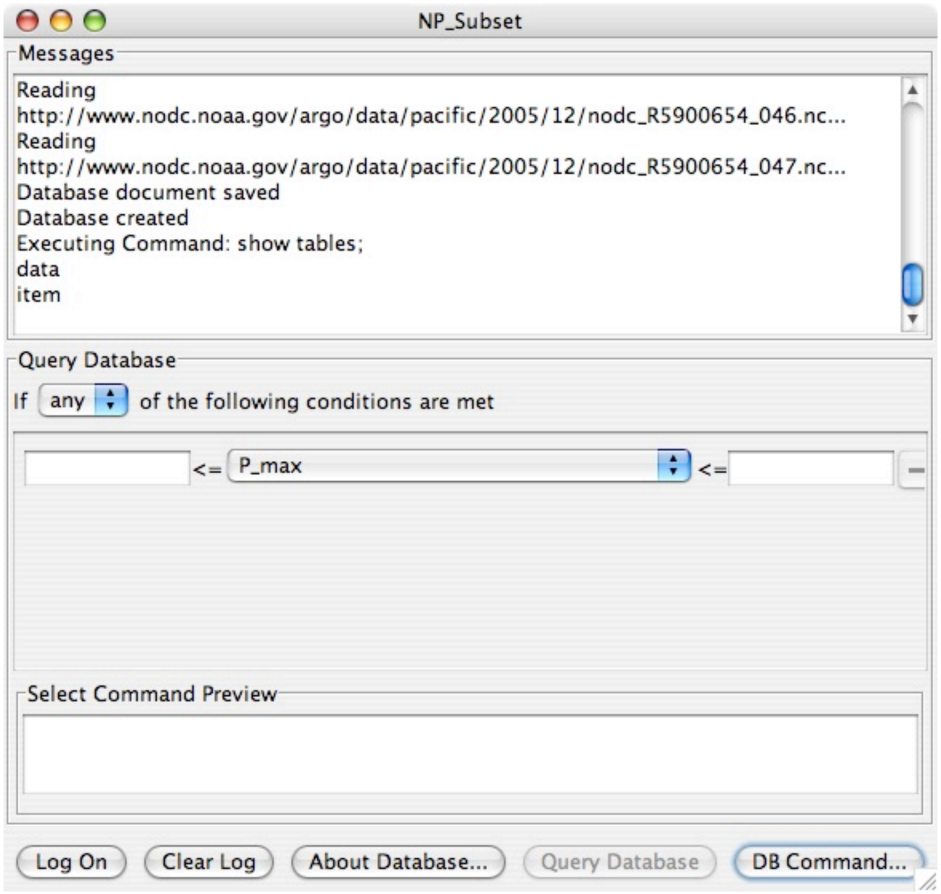
Any MySQL command can be executed directly from NQuery by clicking the *DB Command* button in the database document window. This will display the Enter Command window. Type your command and click the OK button to send you command to the MySQL database server:



Note: All MySQL commands must be terminated by a semicolon.

Viewing Database Tables in the Current Database

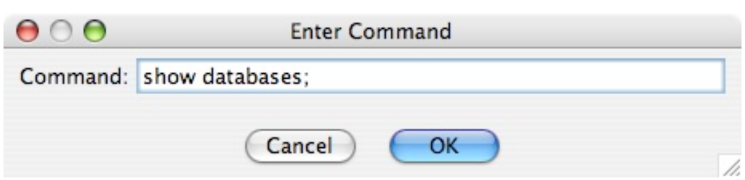
The output of the command will be displayed in the *Messages* area of the Database Document window:

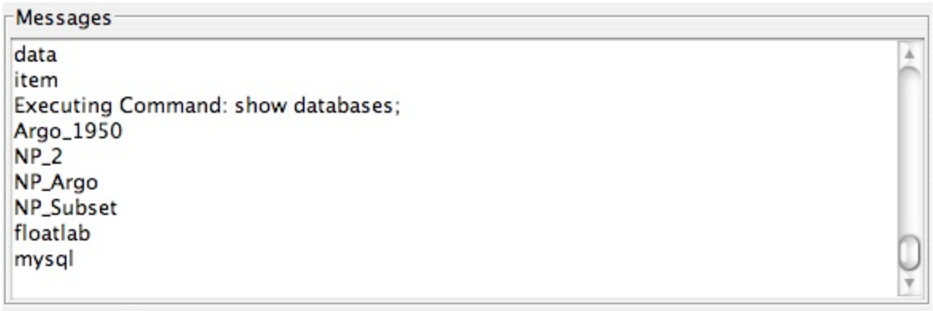


Note: All NQuery databases only contain *data* and *item* tables.

Listing Your Databases

To display a list of all the databases you have created with NQuery, use the MySQL show databases command:



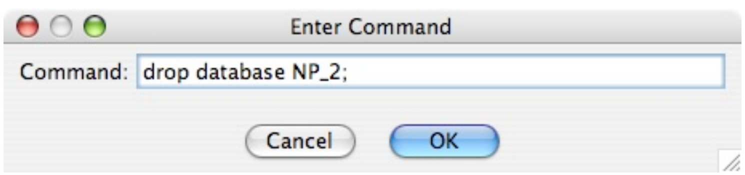


Deleting a Database

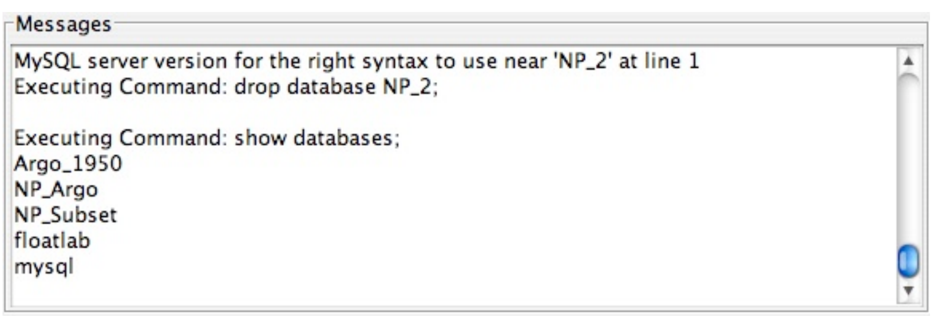
Deleting a database document at the operating system level does not delete the actual database stored in the MySQL server. To delete a database, you need to drop it either through the MySQL command line interface or by issuing a drop command from NQuery. Make sure you delete the associated database document file after you drop its associated database.

1) Use the “show databases” command described above to make sure the database exists (see above).

2) Click the *DB Command* button and enter the *drop* command:

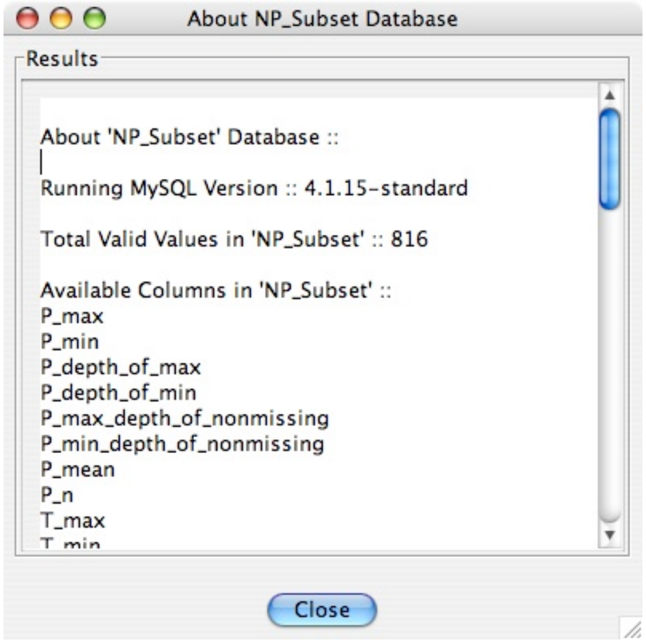


3) You will receive confirmation in *Messages* that the command executed. A subsequent *show databases* command confirms that the database has been dropped:



Getting Information About the Current Database

Click the *About Database* button at the bottom of the *Database Document* to display a summary of the database represented by this document:



This window list the MySQL version, the number of valid values in the database, as well as all the column names.

Other Database Document Functions

Log Off: Disconnects a Database Document from the database server. To use the database, you will have to reconnect using the *Log On* button and providing a valid password.

Clear Log: This button clears the text from Messages area of the Database Document window.

APPENDIX A: ADDITIONAL INFORMATION

Algorithms and Documentation

(adapted from the original document written by E. Flinchem for the User Manual for the Mac OS application Power OceanAtlas)

1. Overview

Java OceanAtlas provides considerable internal facilities for computing new results from input data. Calculations fall into five broad categories:

• Scalar functions
• Scalar functions smoothed in z
• Scalar functions integrated in z
• Interpolations onto fixed z-levels, *i.e.* grids

Results are reported in SI units except where otherwise noted.

2. Scalars

A cast consists of a series of levels ordered by increasing depth at a latitude, longitude coordinate pair. At each level (also known as a 'bottle') there exists a set of observed quantities, such as salinity (*s*), temperature (*t*), and pressure (*p*). Polynomial expansions in *s, t,* and *p* are used to compute the following derived quantities:

• density
• adiabatic lapse rate
• heat capacity
• sound velocity
• NO (respiration corrected nitrate)
• PO (respiration corrected phosphate)
• apparent oxygen utilization

From the elementary polynomials, four more derived quantities are constructed:

• potential temperature at one atmosphere
• potential density - 1000, i.e. sigma-theta, at five reference pressures
• specific volume anomaly
• heat content

The polynomials are taken from the literature and are reproduced here for reference.

NO and PO

JGR, 90, 6925-6939, 1985. Quantities must be in micromoles/liter.

$$NO(NO_3, O_2) = 8.7864 \times NO_3 + O_2$$
$$PO(PO_4, O_2) = 170.8467 \times PO_4 + O_2$$

Apparent Oxygen Utilization

Solubility Data Series v.7.

$$AOU = e^{-1268.9782 + \frac{36063.19}{T} + 220.1832 \times \log(T) - 0.351299\,T + S \times (0.006299 - \frac{3.5912}{T}) + 0.00000344 \times S^2} - O_2$$

Heat Capacity

JGR, 78, 4499-4507, 1973.

$$Cp(T) = 4.2174 - 3.720283 \times 10^{-3}T + 1.412855 \times 10^{-4}T^2 - 2.654387 \times 10^{-6}T^3 + 2.093236 \times 10^{-8}T^4$$
$$A = -1.381 \times 10^{-2} + 1.938 \times 10^{-4}T - 2.5 \times 10^{-6}T^2$$
$$B = 4.3 \times 10^{-4} - 9.9 \times 10^{-6}T + 1.3 \times 10^{-7}T^2$$
$$Cl‰ = \frac{S‰}{1.80655}$$
$$Cp(T, Cl‰) = Cp(T) + A \times Cl + B \times Cl^{3/2}$$

Adiabatic Temperature Gradient

Bryden, *DSR,* 20, 401-408, 1973. P is in decibars.

$$\Gamma(S, T, P) = \sum_i \sum_j \sum_k A_{ijk} P^i (S - 35)^j T^k$$

$$\begin{array}{ll}
A_{000} = \ \ 0.35803 \times 10^{-1} & A_{102} = \ \ 0.87330 \times 10^{-8} \\
A_{001} = \ \ 0.85258 \times 10^{-2} & A_{103} = -0.54481 \times 10^{-10} \\
A_{002} = -0.68360 \times 10^{-4} & A_{110} = -0.11351 \times 10^{-6} \\
A_{003} = \ \ 0.66228 \times 10^{-6} & A_{111} = \ \ 0.27759 \times 10^{-8} \\
A_{010} = \ \ 0.18932 \times 10^{-2} & A_{200} = -0.46206 \times 10^{-9} \\
A_{011} = -0.42393 \times 10^{-4} & A_{201} = \ \ 0.18676 \times 10^{-10} \\
A_{100} = \ \ 0.18741 \times 10^{-4} & A_{202} = -0.21687 \times 10^{-12} \\
A_{101} = -0.67795 \times 10^{-6} &
\end{array}$$

Sound Velocity

Chen and Millero, *J. of the Acoustical Society of America,* 62, 1129-1135, 1977. P is in bars.

$$a_0 = 1.389 - 1.262 \times 10^{-2}t + 7.164 \times 10^{-5}t^2 + 2.006 \times 10^{-6}t^3 - 3.21 \times 10^{-8}t^4$$
$$a_1 = 9.4742 \times 10^{-5} - 1.258 \times 10^{-5}t - 6.4885 \times 10^{-8}t^2 + 1.0507 \times 10^{-8}t^3 - 2.0122 \times 10^{-10}t^4$$
$$a_2 = -3.9064 \times 10^{-7} + 9.1041 \times 10^{-9}t - 1.6002 \times 10^{-10}t^2 + 7.988 \times 10^{-12}t^3$$
$$a_3 = 1.1 \times 10^{-10} + 6.649 \times 10^{-12}t - 3.389 \times 10^{-13}t^2$$
$$a = a_0 + pa_1 + p^2a_2 + p^3a_3$$
$$b_0 = -1.922 \times 10^{-2} - 4.42 \times 10^{-5}t$$
$$b_1 = 7.3637 \times 10^{-5} + 1.7945 \times 10^{-7}t$$
$$b = b_0 + b_1p$$
$$c_0 = 1402.388 + 5.03711t - 5.80852 \times 10^{-2}t^2 + 3.342 \times 10^{-4}t^3 - 1.478 \times 10^{-6}t^4 + 3.1464 \times 10^{-9}t^5$$
$$c_1 = 0.153563 + 6.8982 \times 10^{-4}t - 8.1788 \times 10^{-6}t^2 + 1.3621 \times 10^{-7}t^3 - 6.1185 \times 10^{-10}t^4$$
$$c_2 = 3.126 \times 10^{-5} - 1.7107 \times 10^{-6}t + 2.5974 \times 10^{-8}t^2 - 2.5335 \times 10^{-10}t^3 + 1.0405 \times 10^{-12}t^4$$
$$c_3 = -9.7729 \times 10^{-9} + 3.8504 \times 10^{-10}t - 2.3643 \times 10^{-12}t^2$$
$$c = c_0 + pc_1 + p^2c_2 + p^3c_3$$
$$d = 1.727 \times 10^{-3} - 7.8936 \times 10^{-6}p$$
$$c_{sound} = c + as + bs^{3/2} + ds^2$$

Spiciness (Jacket and McDougall)

Jackett and McDougall, *DSR,* 32A, 1195-1208, 1985.

$$\tau(\theta,S) = \sum_{i=0}^{4} \sum_{j=0}^{4} a_{ij}\, \theta^{i} S^{j}$$

$$a = \begin{bmatrix} 1.609705 \times 10^{-1} & 6.542397 \times 10^{-1} & 5.222258 \times 10^{-4} & -2.586742 \times 10^{-5} & 7.565157 \times 10^{-7} \\ -8.007345 \times 10^{-2} & 5.309506 \times 10^{-3} & -9.612388 \times 10^{-5} & 3.211527 \times 10^{-6} & -4.610513 \times 10^{-8} \\ 1.081912 \times 10^{-2} & -1.561608 \times 10^{-4} & 3.774240 \times 10^{-6} & -1.150394 \times 10^{-7} & 1.146084 \times 10^{-9} \\ -1.451748 \times 10^{-4} & 3.485063 \times 10^{-6} & -1.387056 \times 10^{-7} & 3.737360 \times 10^{-9} & -2.967108 \times 10^{-11} \\ 1.219904 \times 10^{-6} & -3.591075 \times 10^{-8} & 1.953475 \times 10^{-9} & -5.279546 \times 10^{-11} & 4.227375 \times 10^{-13} \end{bmatrix}$$

Spiciness (Flament)

Flament, P., Progress in Oceanography Volume 54, 2002, Pages 493-501

$$\pi(\theta,s) = \sum_{i=0}^{5} \sum_{j=0}^{4} b_{ij}\theta^{i}(s-35)^{j}$$

where:

i	j	b[i,j]
0	0	0.0
0	1	7.7442 (-1)
0	2	-5.85 (-3)
0	3	-9.84 (-4)
0	4	-2.06 (-4)
1	0	5.1655 (-2)
1	1	2.034 (-3)
1	2	-2.742 (-4)
1	3	-8.5 (-6)
1	4	1.36 (-5)
2	0	6.64783 (-3)
2	1	-2.4681 (-4)
2	2	-1.428 (-5)
2	3	3.337 (-5)
2	4	7.894 (-6)
3	0	-5.4023 (-5)
3	1	7.326 (-6)
3	2	7.0036 (-6)
3	3	-3.0412 (-6)
3	4	-1.0853 (-6)

4	0	3.949 (-7)
4	1	-3.029 (-8)
4	2	-3.8209 (-7)
4	3	1.0012 (-7)
4	4	4.7133 (-8)
5	0	-6.36 (-10)
5	1	-1.309 (-9)
5	2	6.048 (-9)
5	3	-1.1409 (-9)
5	4	-6.676 (-10)

Density

Millero, *et al.*, *DSR,* 27A, 255-264, 1980.
Millero, and Poisson *DSR,* 28A, 625-629, 1981. P is in bars.

$$\rho_w^0 = 999.842594 + 6.793952 \times 10^{-2}T - 9.095290 \times 10^{-3}T^2 + 1.001685 \times 10^{-4}T^3 - 1.120083 \times 10^{-6}T^4 + 6.536332 \times 10^{-9}T^5$$

$$A = .824493 - 4.0899 \times 10^{-3}T + 7.6438 \times 10^{-5}T^2 - 8.2467 \times 10^{-7}T^3 + 5.3875 \times 10^{-9}T^4$$

$$B = -5.72466 \times 10^{-3} + 1.0227 \times 10^{-4}T - 1.6546 \times 10^{-6}T^2$$

$$C = 4.8314 \times 10^{-4}$$

$$a = 54.6746 - 0.603459T + 1.09987 \times 10^{-2}T^2 - 6.1670 \times 10^{-5}T^3$$

$$b = 7.944 \times 10^{-2} + 1.6483 \times 10^{-2}T - 5.3009 \times 10^{-4}T^2$$

$$c = 2.2838 \times 10^{-3} - 1.0981 \times 10^{-5}T - 1.6078 \times 10^{-6}T^2$$

$$d = 1.9107 \times 10^{-4}$$

$$e = -9.9348 \times 10^{-7} + 2.0816 \times 10^{-8}T + 9.1697 \times 10^{-10}T^2$$

$$A_w = 3.239908 + 1.43713 \times 10^{-3}T + 1.16092 \times 10^{-4}T^2 - 5.77905 \times 10^{-4}T^3$$

$$B_w = 8.50935 \times 10^{-5} - 6.12293 \times 10^{-6}T + 5.2787 \times 10^{-8}T^2$$

$$K_w^0 = 19652.21 + 148.4206T - 2.327105T^2 + 1.360477 \times 10^{-2}T^3 - 5.155288 \times 10^{-5}T^4$$

$$D = A_w + cS + dS^{3/2}$$

$$E = B_w + eS$$

$$K^0 = K_w^0 + aS + bS^{3/2}$$

$$K = K^0 + DP + EP^2$$

$$\rho^0 = \rho_w^0 + AS + BS^{3/2} + CS^2$$

$$\rho = \frac{\rho^0}{1 - \frac{P}{K}}$$

Other Scalars

Theta is generated from the adiabatic temperature gradient by a single step of Runge-Kutta integration, Fofonoff, *DSR,* 24, 489-491, 1977.

Specific volume anomaly follows directly from density.

Sigma-Theta is formed by substitution of *theta* for *t* in the density polynomial.

Heat Content is the product of the temperature and the specific heat.

3. Scalar functions smoothed in z

Three quantities related to the buoyancy frequency are available, n, n^2, and fn^2/g. The approximation used is:

$$n^2 = \frac{g\frac{d\rho}{dz}}{\rho}$$

The raw densities are smoothed by the application of a Gaussian weighting function, $e^{-(a/z)^2}$, where a and z are in meters, and then centered finite differences are taken. The result is reported in cycles per hour.

4. Scalar functions integrated in z

The following quantities are computed as running sums of trapezoids down each cast:

- Geopotential Anomaly
- Heat Storage
- Potential Energy Anomaly
- Acoustic Travel Time

Sums are computed from the surface down with a sign convention which gives increasing negative values. When a reference level at depth is subtracted, the values above the reference level are typically positive. If the uppermost data point in a cast is not at the surface, then the layer between the surface and the first data point is assumed to have uniform properties. Note the following definition:

$$\Delta p_i \equiv p_i - p_{i+1}$$
$$\Delta z_i \equiv z_{i+1} - z_i$$

Geopotential Anomaly

$$\Delta\Phi = -\int_{p=0}^{p=P_B} \delta_{s,t,p}\,dp$$

$$\Delta\Phi_n = \frac{1}{2}\sum_{i=1}^{n} \left(\delta_i+\delta_{i+1}\right)\Delta p_i$$

Net Heat Content

$$\mathrm{HEAT} = -\int_{z=0}^{z=\mathrm{Z_B}} \rho_\Theta C p_{s,\Theta}\Theta \mathrm{d}z$$

$$\mathrm{HEAT}_n = \frac{1}{8}\sum_{i=1}^{n} (Cp_i+Cp_{i+1})\left(\Theta_i+\Theta_{i+1}\right)(\rho_{\Theta i}+\rho_{\Theta i+1})\,\Delta z_i$$

Potential Energy Anomaly

$$\chi = -\frac{1}{g}\int_{p=0}^{p=\mathrm{P_B}} p\delta_{s,t,p}\mathrm{d}p$$

$$\chi_n = \frac{1}{4g}\sum_{i=1}^{n} \left(\delta_i+\delta_{i+1}\right)(p_i+p_{i+1})\,\Delta p_i$$

Acoustic Travel Time

$$\mathrm{T} = -\int_{z=0}^{z=\mathrm{Z_B}} \frac{\mathrm{d}z}{c_{s,t,p}}$$

$$\mathrm{T}_n = \frac{1}{2}\sum_{i=1}^{n} \frac{\Delta z_i}{(c_i\text{-}c_{i+1})}$$

APPENDIX B: DATABASE SCHEMA

NQuery database have an *item* table for metadata and a *data* table for the results of built-in calculations and station calculations. The *item_id* key is typically constructed from the path or URL of an individual profile.

Here is the schema for the item table:

Key	Column	Data Type	Length	Not Null	Index	Default Value	Extra
☑	item_id	varchar(254)	254	NOT NULL	KEY		
☐	data_type	varchar(254)	254	NULL			
☐	fileset	varchar(254)	254	NULL			
☐	id	varchar(254)	254	NULL			
☐	longitude	double	22	NULL			
☐	latitude	double	22	NULL			
☐	zmin	double	22	NULL			
☐	zmax	double	22	NULL			
☐	zunits	varchar(254)	254	NULL			
☐	date	datetime	19	NULL			
☐	start_time	datetime	19	NULL			
☐	end_time	datetime	19	NULL			
☐	variables	text	65535	NULL			
☐	units	text	65535	NULL			
☐	lexicon	varchar(254)	254	NULL			

Here is the schema for the *data* table a typical NQuery database:

Key	Column	Data Type	Length	Not Null	Index	Default Value	Extra
☑	item_id	varchar(254)	254	NOT NULL	KEY		
☐	P_max	double	22	NULL			
☐	P_min	double	22	NULL			
☐	P_depth_of_max	double	22	NULL			
☐	P_depth_of_min	double	22	NULL			
☐	P_max_depth_of_nonmissi	double	22	NULL			
☐	P_min_depth_of_nonmissi	double	22	NULL			
☐	P_mean	double	22	NULL			
☐	P_n	double	22	NULL			
☐	T_max	double	22	NULL			
☐	T_min	double	22	NULL			
☐	T_depth_of_max	double	22	NULL			
☐	T_depth_of_min	double	22	NULL			
☐	T_max_depth_of_nonmiss	double	22	NULL			
☐	T_min_depth_of_nonmissi	double	22	NULL			
☐	T_mean	double	22	NULL			
☐	T_n	double	22	NULL			
☐	S_max	double	22	NULL			
☐	S_min	double	22	NULL			
☐	S_depth_of_max	double	22	NULL			
☐	S_depth_of_min	double	22	NULL			

Key	Column	Data Type	Length	Not Null	Index	Default Value	Extra
	S_max_depth_of_nonmissi	double	22	NULL			
	S_min_depth_of_nonmissi	double	22	NULL			
	S_mean	double	22	NULL			
	S_n	double	22	NULL			
	THTA_max	double	22	NULL			
	THTA_min	double	22	NULL			
	THTA_depth_of_max	double	22	NULL			
	THTA_depth_of_min	double	22	NULL			
	THTA_max_depth_of_nonr	double	22	NULL			
	THTA_min_depth_of_nonn	double	22	NULL			
	THTA_mean	double	22	NULL			
	THTA_n	double	22	NULL			
	SIG0_max	double	22	NULL			
	SIG0_min	double	22	NULL			
	SIG0_depth_of_max	double	22	NULL			
	SIG0_depth_of_min	double	22	NULL			
	SIG0_max_depth_of_nonm	double	22	NULL			
	SIG0_min_depth_of_nonmi	double	22	NULL			
	SIG0_mean	double	22	NULL			
	SIG0_n	double	22	NULL			
	SPCY_max	double	22	NULL			

Key	Column	Data Type	Length	Not Null	Index	Default Value	Extra
	SIG0_mean	double	22	NULL			
	SIG0_n	double	22	NULL			
	SPCY_max	double	22	NULL			
	SPCY_min	double	22	NULL			
	SPCY_depth_of_max	double	22	NULL			
	SPCY_depth_of_min	double	22	NULL			
	SPCY_max_depth_of_nonn	double	22	NULL			
	SPCY_min_depth_of_nonm	double	22	NULL			
	SPCY_mean	double	22	NULL			
	SPCY_n	double	22	NULL			
	AOU_max	double	22	NULL			
	AOU_min	double	22	NULL			
	AOU_depth_of_max	double	22	NULL			
	AOU_depth_of_min	double	22	NULL			
	AOU_max_depth_of_nonm	double	22	NULL			
	AOU_min_depth_of_nonmi	double	22	NULL			
	AOU_mean	double	22	NULL			
	AOU_n	double	22	NULL			
	T_on_P_at_Surface	double	22	NULL			
	S_wrt_P_between_1000_0_	double	22	NULL			
	MLDF_T_5_05	double	22	NULL			

APPENDIX C: EPIC XML POINTER FILE DTD

```
<!--DTD for EPIC Profile Data -->
<!-- ? =  optional, not repeatable -->
<!-- + =   required and repeatable -->
<!-- * =  optional and repeatable -->

<!ELEMENT epicxml(domain, varlist?, fileset+, attribute*,
comment*)>
<!ATTLIST epicxml version CDATA #REQUIRED
                  type (profile | time-series | grid | other)
#REQUIRED
                  URI CDATA #IMPLIED
                  lexicon CDATA #IMPLIED>

<!-- domain for epicxml must consist of location=”north”,
“south”, “east”, “west”, “top”, “bottom”, “start”,  and “end”
for latitude, longitude, vertical, and time, respectively -->
<!ELEMENT domain(latitude+, longitude+, vertical+, (time | date)+,
attribute*, comment*)>

<!ELEMENT deltat (#PCDATA)>
<!ATTLIST deltat units CDATA #IMPLIED "minutes">     <!--
acceptable units are seconds, minutes, hours, days, unacceptable
units include months, years -->

<!ELEMENT time (#PCDATA)>
<!ATTLIST time location (start | end | point) #REQUIRED “point”
               units CDATA #IMPLIED >

<!ELEMENT date EMPTY>
<!ATTLIST date location (start | end | point) #REQUIRED “point”
                year CDATA #REQUIRED
                month CDATA #REQUIRED
                day CDATA #REQUIRED
                hour CDATA #IMPLIED
                min CDATA #IMPLIED
                secs CDATA #IMPLIED>

<!ELEMENT latitude (#PCDATA)>
<!ATTLIST latitude location (north | south | start | end |
point) #REQUIRED “point”
                    units (degrees_north | degrees_south)
#IMPLIED “degrees_north”>

<!ELEMENT longitude (#PCDATA)>
<!ATTLIST longitude location (east | west | start | end | point)
#REQUIRED “point”
```

```
                    units (degrees_east | degrees_west) #IMPLIED
“degrees_east”>

<!ELEMENT vertical (#PCDATA)>
<!ATTLIST vertical location (top | bottom | start | end | point)
#REQUIRED “point”
                   units CDATA #REQUIRED
                   positive CDATA "down">

<!ELEMENT varlist ((variable | variableref)+, attribute*,
comment*)>
<!ATTLIST varlist lexicon CDATA #IMPLIED>

<!ELEMENT variable (attribute*, comment*)>
<!ATTLIST variable name ID #REQUIRED
                   units CDATA #REQUIRED
                   description CDATA #IMPLIED
                   lexicon CDATA #IMPLIED>

<!ELEMENT variableref EMPTY>
<!ATTLIST variableref refname IDREF #REQUIRED>

<!ELEMENT fileset (varlist?, station*, grid*, track*,
attribute*, comment*)>
<!ATTLIST fileset id CDATA #REQUIRED
                  URI CDATA #IMPLIED>

<!-- time should have location=”start”, “end” for time series,
“point” for profile,
vertical should have location=”top”, “bottom” for profile,
“point” for time series,
latitude and longitude should have location=”point” -->
<!ELEMENT station (varlist?, (time | date)+, latitude,
longitude, vertical+, deltat?, stationvalues*, attribute*,
comment*)>
<!ATTLIST station id CDATA #REQUIRED
                  cast CDATA #IMPLIED
                  URI CDATA #IMPLIED
                  bottom CDATA #IMPLIED
                  reference CDATA #REQUIRED>

<!-- domain for grid must consist of location=”north”, “south”,
“east”, “west”, “top”, “bottom”, “start”,  and “end” for
latitude, longitude, vertical, and time, respectively -->
<!ELEMENT grid (varlist?, domain, attribute*, comment*)>
<!ATTLIST grid id CDATA #REQUIRED
               URI CDATA #IMPLIED
               reference CDATA #REQUIRED>
```

```
<!-- domain for track must consits of location="start" and
location="end" for the latitude, longitude, vertical, and time
elements -->
<!ELEMENT track (varlist?, domain, attribute*, comment*)>
<!ATTLIST track id CDATA #REQUIRED
               URI CDATA #IMPLIED
               reference CDATA #REQUIRED>

<!ELEMENT stationvalue (attribute*, comment*)>
<!ATTLIST stationvalue cast CDATA #REQUIRED
                 name CDATA #REQUIRED
                 units CDATA #REQUIRED
                 method CDATA #IMPLIED
                 lexicon CDATA #IMPLIED
                 value CDATA #REQUIRED>

<!ELEMENT attribute EMPTY>
<!ATTLIST attribute name CDATA #REQUIRED
                    value CDATA #REQUIRED>

<!ELEMENT comment (#PCDATA)>
```

APPENDIX D: REPORTING BUGS TO THE NQUERY DEVELOPER

Mac OS X

NQuery will print any kind of error messages in the System Console. Open the Console application found in the Applications/Utilities folder. If you see something that looks like this...

```
java.lang.Exception
        at javaoceanatlas.PowerOceanAtlas.init(PowerOceanAtlas.java:180)
        at javaoceanatlas.PowerOceanAtlas.<init>(PowerOceanAtlas.java:78)
        at javaoceanatlas.PowerOceanAtlas.main(PowerOceanAtlas.java:533)
```

...then NQuery has encountered a problem. This could either be a bug in NQuery or an issue with a data file. Please copy this text and email to me--having an exception trace is very valuable in tracking down a problem. If possible, provide me with any data files that you think may be causing the problem to appear.

Windows and Linux

If you are having problems with NQuery not doing what you expect it do, it may have encountered an error. It would be very helpful to me for you to enable the console in NQuery and send me any error messages. Do this:

0) Quit NQuery

1) In the folder where the NQuery program is installed, there should be a file called: NQuery 1.0.lax

2) Open this folder into a basic text editor (like Notepad), NOT a word processor like Word.

3) Change these lines:

lax.stderr.redirect=
lax.stdout.redirect=

To:

lax.stderr.redirect=console
lax.stdout.redirect=console

4) Save the .lax file (make sure it's saved in "text only" format)

5) When you restart NQuery, you will see a "command" window. If you see something that looks like this in the command window:

```
java.lang.Exception
	at javaoceanatlas.PowerOceanAtlas.init(PowerOceanAtlas.java:180)
	at javaoceanatlas.PowerOceanAtlas.<init>(PowerOceanAtlas.java:78)
	at javaoceanatlas.PowerOceanAtlas.main(PowerOceanAtlas.java:533)
```

then NQuery has encountered a problem. This could either be a bug in NQuery or an issue with a data file. Please copy this text and email to me--having an exception trace is very valuable in tracking down a problem. If possible, provide me with any data files that you think may be causing the problem to appear.